SCALABLE FRAMEWORK FOR SECURING CONSTRAINED EDGE DEVICES INTO AN INFORMATION-CENTRIC NETWORK OF THINGS

by

Nicholas Clark
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Information Technology

Committee:

_____	Dr. J. Mark Pullen, Dissertation Director
_____	Dr. Paulo Costa, Dissertation Co-Director
_____	Dr. Robert Simon, Committee Member
_____	Dr. Elizabeth White, Committee Member
_____	Dr. Michael Hieb, Committee Member
_____	Dr. Deborah Goodings, Associate Dean
Date:_____	Summer Semester 2022 George Mason University Fairfax, VA

Scalable Framework for Securing Constrained Edge Devices into an Information-Centric Network of Things

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Nicholas Clark
Master of Science
George Mason University, 2010
Bachelor of Science
George Mason University, 2009

Director: J Mark Pullen, Professor Emeritus
Department of Computer Science

Summer Semester 2022
George Mason University
Fairfax, VA

DEDICATION

I dedicate this work in memory of my mom, Paula Clark, and to my dad, Roger Clark. If not for them, I could not have accomplished any of this.

I would also like to dedicate this to my friends, Julie, Leeann, Serda, Cathy, Steph, and James that are like family to me and have always been there with support and encouragement. Also, to my cat Gargamel, who has been there for me in both comfort and as a pleasant distraction.

Finally, I dedicate this to the memory of Dr. E. Bernard White, former associate dean for undergraduate studies, who, without his encouragement and support, I would not have resumed my undergraduate academic career so many years ago.

ACKNOWLEDGEMENTS

I would like to first acknowledge Dr. J. Mark Pullen for many years of mentoring and advice. I am extremely grateful for his thorough editing, encouragement, and pushing me to finish despite my frustrations.

I am also thankful to my committee members Dr. Paulo Costa, Dr. Mike Hieb, Dr. Elizabeth White, and Dr. Robert Simon, for supporting me even on my difficult timeline. I am grateful for the advice, suggestions, proofreading, and guidance to help me through to the end.

TABLE OF CONTENTS

Page
List of Tables viii
List of Figures ix
List of Abbreviations x
Abstract xii
1. Introduction 1
1.1 Information-Centric Networking 3
1.2 The IoT 7
1.3 Security and Scalability in the ICN-IoT 10
1.4 Research Questions 12
1.4 Contributions 13
1.5 Dissertation Structure 14
2. Background & Current Research 16
2.1 Evolution of the IoT 16
2.2 IP-IoT Communications 20
2.2.1 IP-IoT Physical and Data-Link/Media Access Control Layers 22
2.2.2 IP-IoT Network Layer 25
2.2.3 IP-IoT Application Layer 26
2.3 Security Aspects of the IoT 27
2.4 ICNs for IoT 30
2.4.1 IP-IoT vs ICN-IoT 30
2.4.2 ICN Features Suited for IoTs 31
2.5 Summary 34
3. High Level Architecture and Supporting Framework 35
3.1 Introduction and Architectural Overview 35
3.2 Related Works to this Contribution 36
3.3 Architecture Contribution 38
3.4 Smart City Use Case 40
3.5 Framework and Supporting Protocols 42
3.5.1 Framework Overview 42

3.5.2 Encryption and Keys Overview 43
3.5.3 Framework and Protocol Walkthrough 47
3.5.3.1 Stage 1: Discovery & Registration 48
3.5.3.2 Stage 2: Device Authentication 52
3.5.3.3 Stage 3: Secure Forwarding Setup 54
3.5.3.5 Stage 4: Service Discovery 56
3.6 Summary 58
4. Enhancing the ICNoT With Trust-Based Access Control 59
4.1 Introduction and Overview 59
4.1.3 Organization 61
4.2 Background and Relation to Previous Chapter 62
4.3 Related Works to this Chapter 64
4.3.1 Attribute-Based Encryption 66
4.4 Proposed Security and Trust Enhancements Scheme 67
4.4.1 Enhanced System Architecture 67
4.4.2 Enhanced Threat Model and Assumptions 69
4.4.3 Security Management 70
4.4.4 Trust Manager 74
4.5 Summary 77
5. Security Evaluation 79
5.1 Informal Threat-Based Security Evaluation 79
5.2 Formal Security Verification 82
5.3 Formal Security Verification Results 84
5.4 Summary 86
6. Scalability and Efficiency Evaluation 87
6.1 Simulation Configuration 88
6.1.1 Simulation Inputs and Design 89
6.1.2 Simulation Outputs 91
6.2 Simulation Results 91
6.2.1 Simulation Results Introduction 91
6.2.2 Simulation in Fixed Area 92
6.2.3 Simulation in Different Densities 96

6.4 Summary 100
7. Conclusions 101
8. References 103

LIST OF TABLES

Table Page

Table 1. Password Derivation Function Parameters....45
Table 2. Symbols and Descriptions....70
Table 3. Attributes of ICN-IoT Nodes and Trust Attributes....73
Table 4. QoS Trust Properties to Monitor....75

LIST OF FIGURES

Figure Page

Figure 1. Incomplete illustration of the hourglass Internet architecture....2
Figure 2. Network Stack Comparison....4
Figure 3. Named-Data Networking Components....7
Figure 4. Open Standards Reference Model [31]....21
Figure 5. IoT Communications Protocols....22
Figure 6. Network Architecture: ICN-IoT Enclave....39
Figure 7. Stages of a device node joining a network....43
Figure 8. Password Derivation Process....44
Figure 9. Parties involved in device integration....47
Figure 10. New device node sends an interest to discover the network....48
Figure 11. Neighbor sends a request to the SM to register a new device....49
Figure 12. Neighbor DN responds to the discover request....51
Figure 13. New DN sends interest to authenticate to the network....53
Figure 14. New DN defines a forwarding path to its CN and service availability....55
Figure 15. New device node can optionally request access to service.....57
Figure 16. Trust and Security Integration....68
Figure 17. Trust Score Calculation....76
Figure 18. Trust credibility evaluation model....77
Figure 19. Formal Security Verification Results....85
Figure 20. Example of device node distribution....90
Figure 21. Average time for completion of device node integration....93
Figure 22. Percentage of nodes with subtree size....94
Figure 23. Transmission by subtree size....96
Figure 24. Device node integration completion times at different densities....98
Figure 25. Transmission burden by subtree size....99

LIST OF ABBREVIATIONS

IPv6 over Low Power Wireless Personal Area Network 6LoWPAN
Attribute-Based Encryption ..... ABE
Entity Authentication and Key Distribution AKEP2
Automated Validation of Internet Security Protocols and Applications ..... AVISPA
Constraint-Logic-based Attack Searcher ..... CL-AtSe
Constrained Application Protocol ..... CoAP
Constrained Resource Environment ..... CoRE
Ciphertext-Policy Attribute-Based Access Control ..... CP-ABE
Cyber-Physical System ..... CPS
Domain Name System ..... DNS
Datagram Transport Layer Security ..... DTLS
Elliptic Curve Cryptography ..... ECC
Elliptic Curve Cryptography Attribute-Based Encryption ..... ECC-ABE
Electronic Product Code ..... EPC
Extended Unique Identifier EUI-64
Wireless Highway Addressable Remote Transducer Protocol ..... HART
High Level Protocols Specification Language ..... HLPSL
Hypertext Transfer Protocol ..... HTTP
Information-Centric Networking ..... ICN
Information-Centric Network of Things ..... ICNoT
Information-Centric Networking Research Group ..... ICNRG
Institute of Electrical and Electronics Engineers ..... IEEE
Internet of Things ..... IoT
Internet of Vehicles ..... IoV
Internet Protocol ..... IP
Internet Protocol version 6 ..... IPv6
Internet Research Task Force ..... IRTF
International Society of Automation ..... ISA
Low Power Wide Area Networking Protocol Standard ..... LoRaWAN
Local Security Manager ..... LSM
Machine-to-Machine M2M
Media Access Control ..... MAC
Message Authentication Code ..... MAC
Named-Data Networking ..... NDN
Named-Data-Networking Simulator in Network Simulator 3 ..... ndnSIM
Near-Field Communication ..... NFC
Network Forwarding Key ..... NFK
National Institute of Standards and Technology ..... NIST
On-the-fly Model Checker ..... OFMC
Object Name Service ..... ONS

Open Systems Interconnect ..... OSI
Pending Interest Table ..... PIT
Pre-Shared Key ..... PSK
Quality of Service ..... QoS
Quick Response (code) ..... QR
Representational State Transfer ..... REST
Request For Comments ..... RFC
Radio Frequency Identification ..... RFID
Routing Protocol for Low Power and Lossy Networks ..... RPL
Supervisory Control and Data Acquisition ..... SCADA
Transport Control Protocol ..... TCP
Transport Layer Security ..... TLS
User Datagram Protocol ..... UDP
Uniform Resource Locator ..... URL
Wireless Sensor Network ..... WSN

ABSTRACT

SCALABLE FRAMEWORK FOR SECURING CONSTRAINED EDGE DEVICES INTO AN INFORMATION-CENTRIC NETWORK OF THINGS

Nicholas Clark, Ph.D.

George Mason University, 2022

Dissertation Director: Dr. J Mark Pullen

Dissertation Co-Director: Dr. Paulo Costa

This thesis provides a framework with associated implementation support, based on Information-Centric Networking (ICN), to address security, efficiency, and scalability challenges in the Internet of Things (IoT). IoT is an important development that aims to interconnect billions of internet-connected devices and sensors and requires extremely high scalability and comprehensive security. The central premise behind ICN is a fundamental change from host-centric-based communication to content-centric with named-driven networking primitives that natively support multicast, mobility, and content-oriented security. ICN has been advanced as an alternative Future Internet architecture based on scalability required for IoT, but as proposed, cannot meet the security needs of IoT. Most IoT devices are heterogeneous and constrained in their available memory, computational, and energy capabilities. Because these devices are so

numerous and can provide critical sensitive information needed to make real-world decisions, special consideration is required in securing them into an ICN-based IoT.

In this thesis, I present a framework and supporting protocols to address authentication registration, secure forwarding, service authorization and discovery of constrained devices into an ICN based IoT in a way that is efficient and highly scalable. To achieve this, I leverage a mesh network with a hierarchical structure to enhance scalability. The device nodes participating in my architecture are assumed to be constrained, so cryptographic operations are kept to a minimum by using lightweight symmetric encryption functions that rely on unconstrained coordinating nodes, in concert with a security manager service to manage authentication and key distribution. This framework works in four stages for a device to securely join the ICN-IoT. It begins with network discovery and registration, followed by device authentication, secure forwarding setup, and service discovery. When the joining process is completed, the device will be able to fully participate in its local ICN-IoT enclave network securely. The framework works by using established secure cryptographic mechanisms and algorithms applied in a novel and efficient way to an ICN while utilizing the interest/data oriented communication model. A case study in the context of a smart city is used to demonstrate the premise. I extend this with a novel approach to allowing the constrained device nodes to improve their security and general computation abilities through secure collaboration and delegation of heavy computational tasks, such as certain cryptographic functions, through the ICN to less constrained edge device nodes. I propose a scheme to enhance

security against insider threats by deploying trust-based access control based on behavioral monitoring of quality-of-service characteristics of the device nodes over time.

My framework and supporting protocols allow these constrained devices operating in low power lossy networks to securely integrate into an ICN-based IoT in the language and style of ICN communications. I accomplish this by using a hierarchical network architecture that consists of enclave networks existing at the internet edge. These enclave networks incorporate coordinating nodes that facilitate the constrained device nodes using an ICN protocol, which I introduce. I demonstrate the security of the framework and protocols using an informal threat-based evaluation and formal security verification, which is presented. The efficiency and scalability are evaluated based on a simulation model.

1. INTRODUCTION

Many of the current problems with today's Internet are a natural consequence of its architecture, which was designed to address communication needs in a time when the network was needed for sharing resources that were expensive and finite, such as mainframe computers. In the beginning, the Internet's essential function was to forward packets among a small number of machines that remained stationary, with well-defined trust relationships. The fundamental design principles of the Internet made it very simple to link new networks and thus facilitated rapid growth. Concurrent to that growth, there were extraordinary innovations in the services and applications able to run on it and in the development of technologies below the network layer that have emerged. This simplicity can be traced to the hourglass-shaped architecture (shown in Figure 1) into which the Internet protocol stack has evolved, where the Internet Protocol (IP) network layer forms its waist [1].

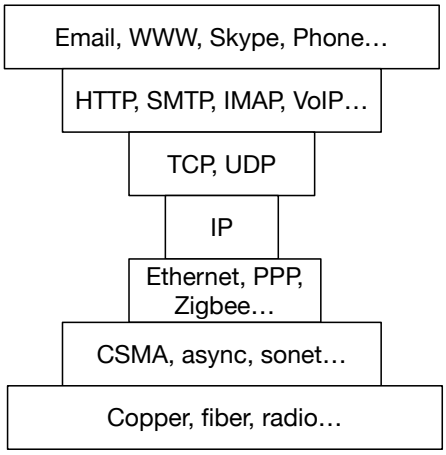

Figure 1. Incomplete illustration of the hourglass Internet architecture

Over time, the growth of the Internet and the introduction therein of new applications to meet emerging needs gave rise to new requirements not met by the existing architecture, such as security, trust, mobility support, scalable content distribution, etc. A cycle of incremental enhancements was introduced to address these new requirements; this continues today. Whenever the Internet faces a new challenge, new patches have been added to address them. Nevertheless, many current and emerging requirements still cannot be addressed adequately by the existing Internet.

According to the Cisco Visual Networking Index forecast, video traffic will compromise 82% of global Internet traffic by 2023 [2]. The majority of this traffic is currently served to end-users via content delivery networks (CDNs), for example, Akamai Technologies, Cloudflare, and Amazon Cloudfront, which are essentially application-layer overlays that cache content on servers near the network edge in order to reduce core network traffic and latency. Cisco expects that by 2023, 71 percent of all Internet traffic will be carried through CDNs [2]. CDNs represent the best current

practice to manage such rapid growth. However, they require substantial infrastructure investment and are often application and provider specific. Despite the benefits to scalability that CDNs have thus far provided, the current host-centric paradigm is not well-suited to scale with the rapid proliferation of mobile devices and the IoT coupled with the rapidly growing volume of video traffic [3].

Such pressing challenges are deeply rooted in the early design decisions of the Internet and may not be entirely solvable without a fundamental rethinking of its architecture. Many of the security problems relate to the Internet's weak notions of identity, such as the ease of spoofing everything from IP addresses to domain names, email addresses, and routing information. Mobility is challenging to handle because IP addresses are hierarchical and tightly coupled with the scalability of routing protocols. Breaking this coupling appears to require a new relationship between naming, addressing, and routing [4]. These issues bring into question whether the approach of continuing to patch over existing patches is the best way to continue or whether a new clean-slate or green-field architectural approach might be needed. Towards this purpose, several research communities that have formed, having identified architectural limitations in the current Internet, are working towards the development of new architectures and paradigms for a Future Internet [5].

1.1 Information-Centric Networking

Information-Centric Networking (ICN) has emerged as one promising candidate for the architecture of the Future Internet. ICN is rooted in the fact that the Internet is

increasingly used for the dissemination of information (content distribution) rather than point to point communication between end hosts. An ICN architecture aspires to better reflect current and future needs better than the existing one by meeting the demands for highly scalable and efficient distribution of content. In an ICN, the networking paradigm is switched from host-centric, where all requests for information are made to a host identified by its IP address(es), to a content-centric one, which decouples named data objects from the hosts that serve them. By naming information at the network layer, ICN favors the deployment of in-network caching and multicast mechanisms, allowing data to be delivered more efficiently and timely to requesting users. Figure 2 shows a comparison of the network stacks between the traditional Open Systems Interconnection (OSI) model, the Transmission Control Protocol & Internet-Protocol (TCP/IP) model, besides the ICN stack. The ICN stack is simplified into three primary layers. The Forwarding layer encompasses the equivalent of routing and forwarding of the network layer, but also in-network storage and caching strategies for the named data.

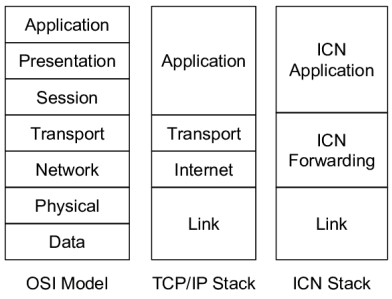

Figure 2. Network Stack Comparison

The named data networking (NDN) project [6] is one ICN architecture that offers capabilities that are useful to the IoT. In the NDN model, there are chunks of data known as content objects that make up a new thin waist in place of the one in Figure 1. Each content object has a unique name, similar to a uniform resource locator (URL) [7]. Naming in NDN uses a hierarchical structure but otherwise arbitrary data identifiers. The content linked to a name is usually considered immutable. To retrieve the desired content object, an interest packet is sent by the requestor to the network. The interest packet must contain at least the name of the desired content object, but it can also include a signature to verify the identity of the requestor. The network retrieves the suitable content object and delivers it back to the requestor in a data packet. The data packet must contain the name, its content, and the signature of the publisher, allowing the requestor to verify the authenticity and integrity of the content object.

Here is an overview of how NDN works: The NDN network includes content routers (CR) that maintain three data structures: pending interest table (PIT), forwarding information base (FIB), and content store (CS). Please see Figure 3 for a visual reference of these components. These tables determine the forwarding procedures for interest and data packets.

When an interest packet is received, the content router first looks in its content store for a match. The CS is a cache of data packets indexed by names. When a match is found in the CS, then the data is served, and the request is considered satisfied. If there is no match, then the PIT is checked. The PIT shows if a previous interest for the same name has already been forwarded and remains unsatisfied. If there is a PIT entry, the CR

doesn't need to forward the interest again. It will instead add the identifier of the face the interest was received on to that PIT record. The term *face* in NDN nomenclature is used similarly to that of a network interface in IP-based nomenclature. It is referenced as a *face* in NDN because it is an abstraction, and while may represent a relation to a physical interface, it can also be tied to other storage or application processes. If there isn't a PIT entry, then the CR consults the FIB, which utilizes a configurable forwarding strategy to identify which face it should forward the interest and update the PIT with a new record that the interest was forwarded.

Data packets are typically forwarded along the reverse path contained in the corresponding PIT entries. When a CR receives a data packet, it checks its PIT to determine the *face* to which it should be forwarded. The PIT record is removed after the data is forwarded, and depending on the caching policies, the data packet may be added to the CS to satisfy future requests. There is a large body of research concerning caching strategies in NDN/ICNs that is outside of the scope of this research.

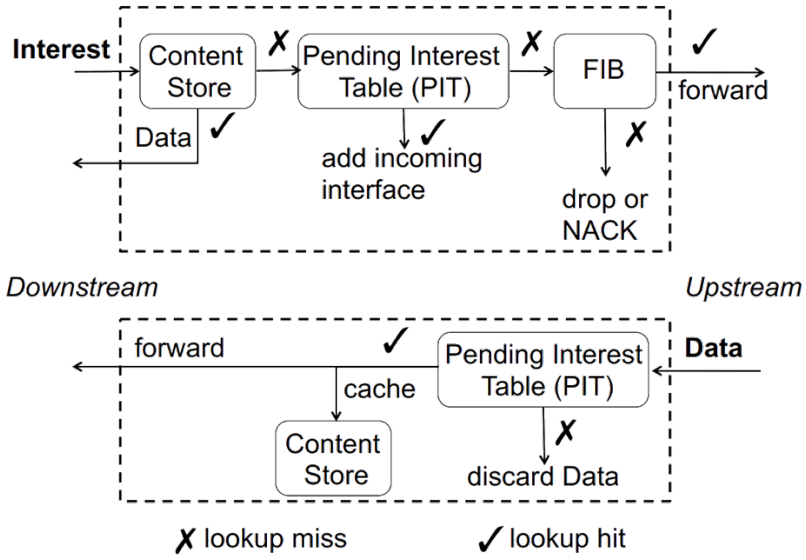

Figure 3. Named-Data Networking Components

1.2 The IoT

The IoT is an emerging paradigm that already involves millions of connected systems and promises to interconnect billions of heterogeneous devices across the Internet. It is facing the challenge of building an infrastructure able to scale and cope with its dynamic environment. The core premise of ICN lies in the name-based routing that enables users to retrieve data objects by names irrespective of their locations. It follows that ICN is well suited for IoT applications, where users consume data generated from IoTs without maintaining secure connections to them. The basic request/response style APIs used in ICN enables developers to build IoT applications simply and efficiently.

Proliferation of low-cost sensing and actuator devices combined with the advancement in wireless communications technologies and the desire to connect and

integrate these devices on a global scale has led to the rise of the IoT, which introduces new risks and challenges. As most IoT devices will be limited in their memory, processing, or energy resources, Internet Engineering Task Force (IETF) working groups have proposed a suite of protocols and open IP-based standards to support IoT-like environments that support these constrained devices. For example, standards *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)*, *IPv6 Routing Protocol for Low power and lossy networks (RPL)*, and *Constrained Application Protocol (CoAP)* are designed to support IP-based IoT solutions. However, significant open challenges remain for deploying IP-based IoT solutions on a large scale [8]. IP-based solutions require a rigid application and device requirements based on the Internet's host-centric IP paradigm. The use of IP addresses implies requirements for an additional resolution system to translate application-level requests, such as in URLs, into IP addresses, connection-oriented end-to-end security, and additional protocols to enable mobility support. Therefore the ICN paradigm has been proposed [8] as an alternative to IP based networking well-suited to the IoT.

As described in section 1.1, in an ICN data is delocalized and does not need to be retrieved via end-to-end transport streams. Rather, hop-wise replication and in-network caching support content dissemination in a way that is well suited to the IoT because it relaxes the demand for constant continued connectivity. ICNs have several attributes that make them well suited to the IoT. In particular, many of the most common communication patterns used in the IoT, such as scheduled and on-request data retrieval,

are easily accommodated in an ICN and could benefit from a cache-assisted, hop-by-hop replication system [9].

In today's IoT, security is a fundamental need. IoT devices and applications often take data from our daily lives through the devices surrounding us. Security in IP-based networks focuses on securing information channels, whereas ICNs perform all security mechanisms on the content itself. In the NDN architecture, each data packet is signed with the original provider's public key and then verified by any consumer in the network allowing the content to be both integrity-assured and authenticated.

An ICN-IoT could reduce the network layers and subsume network, transport, and basic application logic. An ICN approach to IoT could offer opportunities to efficiently factorize core functionalities, such as caching and buffering for error control, reduce the complexity of autoconfiguration mechanisms as compared to those used in a layered protocol stack, and achieve a reduced memory and storage footprint compared to IP based standards and protocols, such as IPv6, 6LoWPAN, and RPL. Some research [10] has validated the applicability of an ICN in the IoT experimentally and demonstrated that it has advantages over IP-based technologies in terms of energy consumption and memory footprint. However, for this to happen several issues remain related to security and scalability. ICNs are an active research area with several working groups, such as the Internet Research Task Force’s (IRTF) Information-Centric Networking Research Group (ICNRG) and the Information-Centric Networking Program National Institute of the Standards and Technology (NIST).

1.3 Security and Scalability in the ICN-IoT

Security is a fundamental requirement and a primary consideration in the design of future Internet and Internet of Things (IoT) applications. This thesis is motivated by my concern with scaling IoT security to the scope of a smart city [11]. The devices and things with which we interact within our lives provide data that often is processed by third parties through cloud service providers. The involvement of these third parties introduces privacy considerations. As described above, in the current IP based Internet security capabilities such as content integrity and device authentication are afterthoughts. Securing the content of data ensures that it remains protected independent of the communication channel.

Protocols such as Transport Layer Security (TLS), Extensible Authentication Protocol (EAP), Protocol for Carrying Authentication for Network Access (PANA), and IPv6-based security solutions rely on securing nodes based on location and the communication channels rather than the content. Resource-constrained IoT devices incur additional delays by adding security mechanisms over IP. Any system, including the IoT, is wholly secured only when it ensures authentication, authorization, confidentiality, and integrity.

The IoT has evolved to include the emerging concepts of smart homes, buildings, campuses, and cities with increasing complexities in requirements for security and scalability. Smart cities, in particular, will contain a large number of devices in various environments with potentially a very high density of deployment [11]. They may involve individual homes, utility services, and other critical infrastructure. The IoT devices

deployed may be resource-constrained with limited computational power, memory, and energy communicating over low-power lossy wireless networks. In the context of these wireless networks, *lossy* is used to refer to the reality that packets may be frequently lost due to a variety of factors such as signal quality or interference. Accommodating this scenario requires new advances in network and communication protocols in scalability and efficiency because these networked devices may also be used to transmit large amounts of sensitive data and manipulate critical infrastructure. Therefore, smart city IoT infrastructure demands strong security and privacy considerations to protect that data and secure the broad attack surface they present.

The large number of IoT devices lacking a user interface can complicate interconnecting them. This is particularly the case with many resource-constrained devices. Various approaches to interconnection have been proposed, based on both conventional network architectures and spontaneous wireless network paradigms [12]. Some of these technologies support the auto-configuration of devices and dynamic self-organization, allowing data to be relayed to the destination without the aid of already configured access points or additional infrastructure. Connectivity of the devices in these environments currently is accomplished through two different categories of network stacks: through proprietary network stacks, such as ZigBee [13], or through open protocol stacks such as IPv6 with 6LoWPAN [14] and RPL [15]. In many cases this is inefficient because proprietary stacks require additional processing in order to translate their communications with the Internet, while deploying the full IP stack has shown to be less resource-efficient [10].

In light of the all this, Information-Centric Networking (ICN) has been shown in recent work to be a more suitable solution for the IoT than IP [10]. An ICN approach to IoT offers opportunities to:

- efficiently factorize core functionalities, such as caching and buffering for error control
- reduce the complexity of auto-configuration mechanisms as compared to those used in a layered protocol stack
- achieve a reduced memory and storage footprint compared to IP-based standards and protocols, such as IPv6 with 6LoWPAN or RPL.

The Named Data Networking (NDN) project [6] is one ICN architecture being widely used with potential for use in larger-scale IoT applications, such as smart cities. My work addresses an approach to securing the IoT by exploiting the capabilities of ICNs using communications entirely in ICN style and format.

1.4 Research Questions

The central premise of my work is described as research questions:

Primary Question:

- How can constrained devices, which make up the vast majority of the Internet of Things, fully integrate into an Information-Centric Network-based Internet of Things in a way that is highly secure, efficient, and scalable?

Secondary Questions:

- How can the security, scalability, and efficiency of a proposed solution be evaluated?

1.4 Contributions

This section describes the contributions of this work and the supporting publications.

- I introduce a framework and supporting protocols for securely integrating a constrained device node into an Information-Centric Network of Things that allows the device to authenticate the network, network to authenticate the device, establish a secure forwarding path, and service authorization and discovery.
- I present a scheme to enable trust-based access control and improved authentication to allow the system to adjust to changes in the behavior of device nodes based on quality-of-service attributes monitored over time.
- I present a simulation to evaluate the scalability and efficiency of the proposed work.
- I evaluate in depth the characteristics and attributes of Information-Centric Network (ICN) that make them well suited for use in the Internet of Things (IoT).

]

Below is a list of published works related to this dissertation.

N. Clark, “ICN Suitability for the Internet of Things,” in *IADIS International Conference e-Society 2020*, 2020, pp. 167–170.

N. K. Clark, “Securely & Efficiently Integrating Constrained Devices into an ICN-IoT,” in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, Jun. 2021, pp. 536–541.

N. K. Clark, “Enhancing an Information-Centric Network of Things at the Internet Edge with Trust-Based Access Control,” in *2022 IEEE 7th World Forum on Internet of Things (WF-IoT)*, Sep. 2022 (Submitted)

1.5 Dissertation Structure

Chapter 1 of this dissertation provides context and general information on Information-Centric Networking (ICN), the Internet of Things (IoT), and related security and scalability challenges. It also provides a high-level motivation for the research and defines the contributions and structure of this work. Chapter 2 provides relevant background information on the evolution of IP-based IoT communications and relevant security aspects to current research. Additionally, it evaluates and compares IP-based IoT to ICN IoT implications and solutions. It describes in detail the features of ICN that make them suited for use in the IoT and evaluates some current research in ICN-based IoT architectures.

Chapter 3 introduces my proposed architecture and supporting protocols for securely integrating a constrained device into an ICN based IoT. It includes an architectural overview, additional comparisons to relevant current research, a description of cryptographic primitives used, and a walkthrough of the protocols.

Chapter 4 provides enhancements to the architecture and protocols in Chapter 3 by extending them through a scheme utilizing trust-based access control to allow the system to dynamically adjust security and access control based on quality-of-service (QoS) attributes monitored on the system.

Chapter 5 provides a security evaluation of my work, beginning with an qualitative threat-based evaluation. It then describes a formal security verification using high-level protocol specification language (HLPSL).

Chapter 6 provides an evaluation of the scalability and efficiency of my work using a unique simulation model developed for this thesis.

Chapter 7 provides a concluding overview of the work and also describes future work to further enhance my architecture and supporting protocols and provide for additional evaluation of its security efficacy and efficiency.

2. BACKGROUND & CURRENT RESEARCH

In this chapter, I discuss background information and existing research and the evolution of existing IP-based IoT communication and security. I further survey existing research to present the state of the art in ICN-based IoT security and focus on specific works most relevant to my work presented in this thesis.

2.1 Evolution of the IoT

The term "Internet of Things" (IoT) is believed to have been first coined in 1999 by Kevin Auston while he was the Executive Director of Auto-ID Labs at MIT, in a presentation to Proctor & Gamble promoting the idea of using Radio-Frequency Identifiers (RFID) for tracking items in their supply chain [16]. The IoT concept was further driven by research in the RFID community focused on linking sensor networks using a convergence of technologies to continually track physical items all over the Earth [17]. This was enabled by different tagging technologies such as Near-Field Communications (NFC), RFID, and 2D barcodes, which permitted physical objects to be identified and referenced over the Internet [18].

Some of the early research into the IoT sought to facilitate the unique tagging, naming, and addressing of these physical objects and the representation and storage of exchanged information. The Electronic Product Code (EPC) was designed as a universal identifier meant to enumerate and identify all objects, for all time. The Object Name Service (ONS) was designed to act like the Domain Name Service (DNS) but for looking

up and resolving object addresses based on EPC numbers instead of domain names [19]. These standards were primarily designed to improve object visibility and traceability with regard to their status, current location, etc., but were important for future capabilities of the IoT.

While work on RFID-based IoT was progressing, research into remote sensing systems utilizing Wireless Sensor Networks (WSN), Telemetry, and Supervisory Control and Data Acquisition (SCADA) technologies for industrial processes also was maturing [20]. Wireless sensor networks consist of a number of physical devices that are deployed in a monitored area in geographic proximity to one another where they communicate together using a wireless multi-hop routing algorithm. The devices create a wireless infrastructure with the purpose of detecting events that occur within the monitored area and conveying that data to one or more dedicated gateways, also referred to as sink nodes or base stations, which eventually transmit the aggregated data to remote management units or servers [21]. WSNs are widely deployed in a wide array of applications for environmental, health, military, smart-home, and industrial purposes.

The sensor node components of the WSN are typically small and inexpensive devices that are often powered using batteries, which makes them highly constrained in capabilities that require energy such as computational power and wireless emissions. The integration of sensors/actuators, RFID tags, and communication technologies serves as the foundation of IoT and explains how a variety of physical objects and devices can be associated to the Internet and allow these objects and devices to communicate and coordinate towards a common goal [22].

The next step in the IoT evolution, Machine to machine (M2M) communications, largely extends the sensor networking model, providing a more advanced type of network with data communication between physical devices without human intervention. M2M networks inherit the resource constrained and mass-deployed nature of sensor networks while enhancing it with embedded intelligence and self-organization [23].

The M2M model can be characterized by three primary properties. The first is diversity - it has a highly diversified pool of components, ranging from low-resource sensors to powerful servers, that may be distributed throughout a large geographical area. The second is the autonomy of the component's operations compared with the legacy Internet. While M2M systems are designed to be decentralized and minimize the requirement for human involvement, the more advanced ones may implement functions of situational awareness, self-organization, or cognition [24]. Finally, M2M systems incorporate a distributed communication model in which any two nodes may establish a relationship with each other if one of the nodes is offering a service or resource which is required of the other.

The development of M2M systems diverges from the logical and topological simplicity of sensor networks. Sensor nodes in an M2M environment might not simply interact along hierarchical direct paths such as between the sensor, sink nodes, and remote management units. A sensor in an M2M environment is likely to have direct interactions with other peers of varying distance and capabilities based on the desired services or resources that it needs to exchange [23]. This paradigm where a heterogeneous collection of nodes interact through a decentralized communication

pattern leads to situations where there may be imbalances in computational and energy resources between interacting peers. These imbalances impact, among other things, the ability of the peers to utilize security mechanisms involving cryptography and complex algorithms.

The development of RFIDs, WSNs, pervasive computing technologies, and M2M systems, combined with advanced network communication and emerging control theory, cyber-physical systems (CPS), has emerged as a new pattern in the IoT. CPS is an evolution of WSN and M2M where multiple dimensions of sensing data, crossing multiple sensor networks and the Internet, emphasizes real-time control functions and aims at constructing intelligence across multiple domains [25]. Examples of CPS systems include cyber-transportation systems with unmanned vehicles with intelligent roads [26] and smart-grid systems with advanced configurability, reactiveness, and self-manageability for next generation electric grids [27].

Research and development continues to evolve the IoT, aiming to interconnect a much wider set of objects, even those that were not natively intended to be able to communicate. For example, barcodes and RFID tags allow otherwise inert objects to advertise their presence and sometimes to receive and store information. This integrates them into the connected world. The advantages of interconnecting huge sets of "things" belong to the field of adaptation, with the ability to sense and respond to the environment, and the field of autonomous orchestration of new services where entities discover one another along with their needs and services. In this perspective, the IoT is defined as a "dynamic global network infrastructure with self-configuring capabilities based on

standard and interoperable communication protocols where physical and virtual ‘things’ have physical attributes, virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network” [28].

2.2 IP-IoT Communications

Several standards and protocols can be used to facilitate communications over the Internet. Figure 4 shows the Open Standards Interconnect (OSI) Reference Model protocol stack applied to a mix of common communication and networking protocols from various standards bodies, illustrating the hierarchy of encapsulation for a range of protocols. Many of these may be used in IoT devices and applications. However, as previously discussed, the nature of many IoT devices, requires them to be extremely power efficient so that they can be powered by batteries or through energy harvesting [29]. Energy is wasted in the transmission of unneeded data, protocol overhead, and in poorly optimized communication patterns. The energy efficiency of transmission is an important consideration when connecting devices to the IoT. Internet Protocols such as Hypertext Transfer Protocol (HTTP) and Transmission Control Protocol (TCP) [30] are not optimized for very low-power communication due to the inclusion of verbose meta-data and headers as well as the requirements for reliability through packet acknowledgement at higher layers.

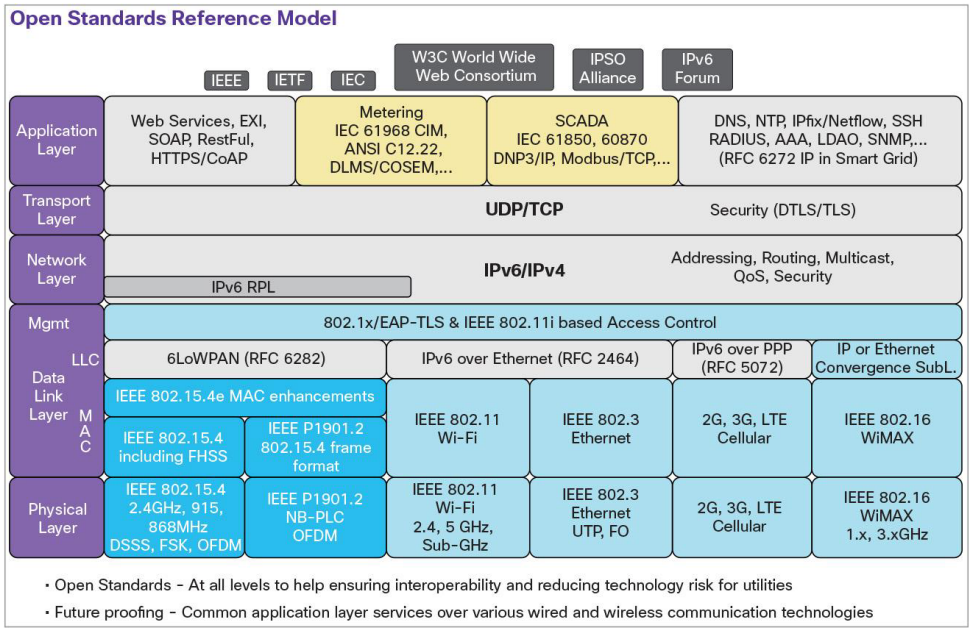

Figure 4. Open Standards Reference Model [31]

Considering these energy constraints and the scale factors of the IoT, many of the communications and security solutions employed in the Internet are poorly suited for the IoT. Working groups formed at standardization bodies, the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF), are designing new communications and security protocols that will play a fundamental role in enabling future IoT applications. These technologies are being designed in line with the constraints and characteristics of low-energy sensing devices and low-rate wireless communications. The protocols already available or still in development at IEEE and IETF have allowed for a standardized protocol stack illustrated in Figure 5 [29]. The protocols forming this stack are designed to enable Internet communications using

resource-constrained devices while fulfilling the requirements for low-energy communication environments.

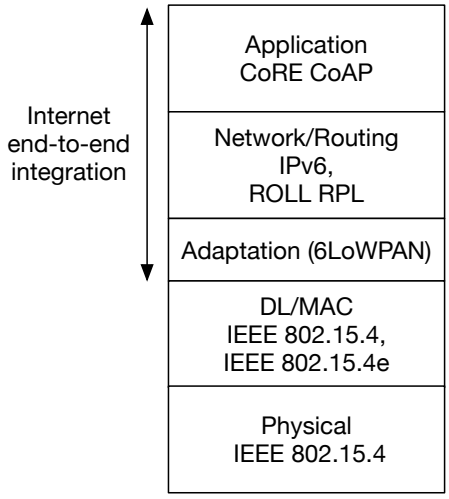

Figure 5. IoT Communications Protocols

The next few sections will evaluate the properties of this IoT network stack including the implications for security.

2.2.1 IP-IoT Physical and Data-Link/Media Access Control Layers

In the IoT stack shown in Figure 5, low-energy communications are supported by IEEE standard 802.15.4, which sets the rules for communications at the lower levels of the networking stack: the physical [29] and media access control (MAC) [33] layers. They lay the foundation for IoT communications at higher levels.

IEEE 802.15.4 supports communications at 250 Kbit/s in a short range of around 10 meters. The original IEEE 802.15.4 standard from 2006 was updated in 2011 to include an evaluation of practical deployments of the standard already in the marketplace. Other amendments were introduced to the standard, namely IEEE 802.15.4a [32], specifying additional physical layers, and IEEE 802.15.4c [34] to support new frequency bands in China and IEEE 802.15.4d [35] for Japan. IEEE 802.15.4e [33], an amendment defining modifications to the MAC layer to support time-synchronized multi-hop communications, is especially relevant for the IoT environment. There are several other amendments covering the use of RFID and smart utility networks.

Because of the suitability for low-energy wireless communication environments, IEEE 802.15.4 creates the foundation for the design of higher level standardized technologies such as 6LoWPAN and CoAP, which are described below. It also has been adopted as the basis of commercial and industrial standards such as ZigBee-2006 [36], ZigBee Pro-2007 [37], ZigBee Pro-2015 [13], International Society of Automation (ISA) 100.11a [38, p. 100], and Wireless Highway Addressable Remote Transducer Protocol (HART) [39]. These technologies are widely deployed in commercial and industrial products, but they were not designed to enable Internet communications from devices. The ZigBee standard defines application profiles that target market areas such as home automation and smart energy systems, while WirelessHART and ISA 100.11a target industrial automation and control markets. There are also other proprietary protocols not directly based on 802.15.4 such as Z-Wave [40], which is primarily used in smart home applications.

In 802.15.4, security mechanisms are only implemented at the MAC layer, below application control. As a link layer security protocol, it provides four basic security services: access control, message integrity, message confidentiality, and replay protection.

- Access control: the link layer protocol prevents unauthorized parties from participating in the network. A legitimate node should be able to detect messages sent from an unauthorized node and reject them.
- Message integrity: A node should be able to detect if a message from a legitimate sender has been altered in transit
- Confidentiality: A node can keep information in a message secret from unauthorized parties. This is typically accomplished using encryption. Ideally, the encryption algorithm should not only keep the message secret, but also prevent an adversary from learning partial information about the message. This property is known as the semantic security [41].
- Replay Protection: This protection prevents an adversary from eavesdropping on a legitimate message between two authorized nodes and then resending the message to a node again at a later time (a replay attack). This is usually prevented by some type of message sequence number inside of the message that is incremented with each new message.

IEEE 802.15.4 provides both encryption and integrity verification which is achieved by a single pre-shared key used for symmetric cryptography. Integrity is

provided by using Message Authentication Codes (MAC) in the packets. The main disadvantage to this approach is that it can only provide security on a per link or hop-to-hop basis, which implies each node must be a trusted entity for the network to be secure.

2.2.2 IP-IoT Network Layer

Internet Protocol Version 6 (IPv6) Over Low-Power Wireless Personal Area Networks (6LoWPAN) is an adaptation layer designed to efficiently encapsulate IPv6 long headers in IEEE802.15.4 small packets, which are limited to a maximum of 128 bytes. The standard supports variable-length addresses, low medium capacity, low power consumption, scalable networks, mobility, unreliability and long sleep time. The standard also provides for header compression to reduce transmission overhead, fragmentation to meet the 128-byte maximum frame length, and support of multi-hop delivery [42].

There are no security mechanisms available in the context of the 6LoWPAN adaptation layer although there are some potential security vulnerabilities and requirements noted in related documentation. Internet standard RFC 4944 considers the possibility of duplicate EUI-64 interface addresses, which are intended to be unique [43]. RFC 6282 discusses security issues that are created because of the problems introduced in RFC 4944 [44]. RFC 6568 addresses potential mechanisms to adopt security within constrained wireless sensor devices [45].

Routing Protocol for Low-Power and Lossy Networks (RPL) is distance-vector protocol in that it relies on distance or hop count as the key metric to determine the best network forwarding path, which supports routing for a variety of datalink protocols. RPL

is designed to work with networks of resource constrained devices. RPL is capable of different levels of security using a special security field after the 4-byte ICMPv6 message header. The contents of this field indicate the security level and cryptography algorithm used for message encryption. RPL provides support for replay protection, data authenticity, semantic security, confidentiality and key management [15].

2.2.3 IP-IoT Application Layer

The Constrained Application Protocol (CoAP) was developed by the Constrained Resource Environment (CoRE) group of the IETF. It is a document transfer protocol designed to provide a lightweight RESTful HTTP interface. Representational State Transfer (REST) is a standard interface used for interaction between HTTP clients and servers. However, CoAP packets are much smaller than those of HTTP. CoAP is specialized for use by constrained devices such as those found in the IoT [46]. It provides a request/response communication model between application level endpoints, supports built-in discovery of services and resources, and includes core concepts of the Web including uniform resource identifiers (URIs) and Internet media types. CoAP is implemented using UDP rather than TCP, which reduces overhead [46].

Because CoAP is built using UDP rather than TCP, the dominant Internet security protocols of Secure Socket Layers (SSL) and Transport Layer Security (TLS) are not applicable. Instead, Datagram Transport Layer Security (DTLS) protocol provides many of the same capabilities of TLS but is available for data sent over UDP datagram protocols. The protocol allows client/server applications to communicate in a way that is

designed to prevent eavesdropping, tampering, or message forgery [47]. The introduction of the IETF CoAP protocol family provides a useful capability in ensuring that traditional Internet applications do not have to be re-engineered in order to run on low-power embedded networks. This approach allows for the same design principles that are currently used general Internet application design to be applied to the IoT.

2.3 Security Aspects of the IoT

- **Confidentiality**: is used to ensure that the data is only available to authorized users throughout its lifecycle and that it cannot be eavesdropped or intercepted by users that have not been authorized to do so. This is most often accomplished through encryption. In the IoT, confidentiality is an important security principle, because many measurement devices, such as RFIDs, sensors, etc., may be integrated into the IoT. It is critical to ensure that the data collected by these devices will not reveal sensitive information.
- **Integrity**: ensures that the data cannot be tampered with by intended or unintended interference during its communication over networks or ultimate access by authorized users or consuming applications.
- **Availability**: ensures that the data and devices are available for authorized users and services whenever they are requested. In IoT, services are commonly requested in real time. Services cannot be scheduled and provided if the requested data cannot be delivered in a timely manner. A common threat to availability is the denial-of-service (DoS) attack where network components are overloaded by adversarial traffic.

- **Identification and Authentication**: Identification ensures that devices or applications that are not authorized cannot be connected to the network. Authentication can ensure that the data delivered in networks are legitimate, and the devices or applications that request the data are legitimate as well. In IoT, identifying and authenticating each data and object is difficult, because many diverse objects comprise an IoT. Thus, designing efficient mechanisms to deal with the authentication of objects or things is critical.
- **Privacy**: is a critical component of the IoT because of the presence of data that might be personal or sensitive. Privacy ensures that the data can only be controlled by the corresponding user, and that no other user can access or process the data.
- **Trust**: is important to ensure the security and privacy objectives to be achieved during the interactions among different objects, IoT layers, and applications. The objectives of trust in IoT can be categorized as trust between each IoT layer, trust between devices, and trust between devices and applications

IoT security is challenging because of the limited resources of most devices, the diversity of those devices, and the massive scale of IoT. Many IoT devices gather and distribute data that is sensitive or personal, which needs to be relayed to different cloud services for processing. The authors of [48] proposed a method to use safe or aggregated answers so as to send as little data as possible to a service provider. In certain scenarios, their methodology may introduce noise to the data to increase privacy, which in some circumstances may also lead to inaccurate services. In [49], user privacy is protected by defining different privacy zones for different types of data. Context based policy

checking functions are assigned to each zone, which are verified by a "Home Security Hub" prior to accepting join or rejoin requests to protect user data against unauthorized sharing. That proposed system does not address the possibility of accessing smart devices directly bypassing the hub or the scalability of the solution. In [50], a methodology using Internet security protocol IPsec is explored to provide authentication and privacy for IoT devices; however, the proposed protocol has limited scalability due to complexities of key management.

The authors of [51] propose a privacy preserving system with three modules for managing data in the smart home: a data collector module that collects users' data, a data receiver that receives data from the collector to be stored in two different datasets, and a result module that controls user access to data in order to protect privacy. One of the datasets includes de-identified sensor data which stores the actual data with primary/quasi- identifiers values hashed. The other is the identifier dictionary storage contains only the hashed and actual values for each unique set of primary/quasi-identifiers, if they do not already exist. The design of the two datasets ensures that linking different data of a user to another is impossible. The system does not provide a way to retain privacy such that it can be shared selectively or to a cloud provider outside of the system.

A distributed capability-based access control protocol, introduced in [52], has several useful qualities for the IoT in that it is designed to be computationally efficient using an optimized implementation of Elliptic Curve Digital Signature Algorithm (ECDSA) and run in a distributed IoT environment. However, the proposed solution

introduces considerable latency in communications. It also does not define how the issuer of the capability tokens used as the foundation for the protocol are initially authorized. Aspects of the presented protocol, such as the ECDSA and its token specifications, may be adaptable for use in other situations.

2.4 ICNs for IoT

This section gives a brief introduction of the concepts related to ICNs and their suitability for IoT. It then discusses the features of ICNs that support IoT and maps them against the architectural requirements of the IoT. It concludes by briefly discussing ICN-based IoT architectures specifically with regard aspects concerning security and scalability.

2.4.1 IP-IoT vs ICN-IoT

TCP/IP originally was engineered to connect a finite number of computers that share an expensive and limited amount of network resources within a limited amount of address space in the network layer. None of these holds true for IoT, so despite its dominance in today's Internet, it is fundamentally flawed in meeting the requirements of IoTs efficiently at scale. Also, the IoT's extensive data usage puts new requirements on the existing IP-based internet architecture involving security, mobility, data dissemination, and scalability.

A flash crowd [53] is a phenomenon that occurs when a large number of Internet users make a request for the same information or resource. As a result of a flash crowd,

there is an increase in network traffic for the host server providing this information and along the network paths servicing to that host, creating congestion. ICN architectures can minimize flash crowds because they support in-network caching. This causes the popular information to be cached in the routers and forwarding devices along the network path. The intermediate routers can send the desired data on behalf of the original producing host minimizing its load. As a result of this capability, the users don't actually need to interact directly with the producing host. They only need to know the name of the information they want so the network itself can provide it.

In a native ICN architecture, the information or content is named independently of its location. Because of this, it can be located anywhere. Naming the data and devices makes the ICN better suited for the IoT because it removes constraints placed by network addressing. An IoT receiver of data is more interested in that data than its location. Content is found on the network based on its location-independent name, and the communication between the producing host and receiver is opaque and more secure.

2.4.2 ICN Features Suited for IoTs

This section discusses the main features of the proposed ICN architectures that make them optimal for IoT.

- Named Data: This is the most prominent feature of ICNs that allows information to be accessed independent of its location. It is a core concept that enables other important features such as in-network caching. Unique names can be assigned to IoT devices, services, or contexts.

- Security: Security and privacy are achieved by protecting the confidentiality of the data rather than just the communication channels as in IP-based networks. The data transmitted between IoT devices and services can be protected even when the communications channels are compromised.
- Resource Efficiency: ICN-based IoTs can offer resource-efficient networking in that data is only provided in response to interests and cached on devices that aren't constrained, reducing the impact on constrained devices.
- Scalability: Name resolution is performed at the network layer with the name state distributed within the entire network. This allows the ICN to achieve high scalability by utilizing features such as local computing, content locality, and multicasting.
- Context-aware communication: ICN-based IoTs allow support for different contexts at different layers, such as the application layer, network layer, and device layer. Contexts at the application layer may be defined be individual higher-level applications; contexts at the network layer include network status and statistics; contexts at the device layer may include information such as energy levels or location. Device context may be available to the network layer, while network entities are able to resolve application layer contexts to contexts at lower layers. Because of this, communications may only occur under conditions that are specified by applications, which can significantly reduce the volume of network traffic.

- Mobility: ICN-based IoT name resolution layer allows multiple levels of mobility. They can rely on the receiver-oriented design for self-recovery of consumers and the ability to use multicasting and late-binding techniques to realize seamless mobility support of the nodes producing data.
- In-network Caching and Storage: This is a key capability of ICNs in general and especially important for ICN-based IoTs. Data is stored locally, either by device nodes, gateway nodes, or routing nodes and at service points or possibly anywhere in the network path. In-network caching speeds up data delivery, especially for 'popular' data, and allows for local repair over unreliable network segments such as those common in the IoT with low-power lossy wireless networks.
- In-network processing: ICN-based IoTs allow for in-network processing for services, such as name and context resolution, data aggregation, and compression.
- Communication reliability: ICN-based IoTs support delay-tolerant communication [54], which allows them to support reliable communication over unreliable network links. Additionally, opportunistic caching provides capabilities to increase the copies of content in the network in response to diverse application and service requirements to address different mobility scenarios.

2.5 Summary

In this chapter, I discuss the background and current research on the Internet of Things and Information-Centric Networking. I began with a walkthrough of the evolution of the Internet of Things and the current IP-based communications technologies and protocols at the physical, data-link, network, and application layers and their impacts on security and performance. I described the most relevant security aspects of the IoT. Finally, I introduced the Information-Centric Networking aspects that directly benefit the IoT and compare those features to the existing IP-based IoT approaches.

3. HIGH-LEVEL ARCHITECTURE AND SUPPORTING FRAMEWORK

3.1 Introduction and Architectural Overview

Previous chapters have described the value of Information-Centric Networking (ICN) for the IoT and highlighted existing security weaknesses of IC for this purpose. This chapter provides an innovative contribution: an architecture based on ICN to address security and scalability challenges in the Internet of Things (IoT). I present a framework and supporting protocols that extend prior work to address authentication, registration, secure forwarding, and service authorization and discovery of constrained devices into an ICN-based IoT in a highly scalable way. This approach allows constrained devices operating in low-power lossy networks to achieve required security using ICN communication style and format. The device nodes participating in our architecture are assumed to be resource-constrained, so cryptographic operations are kept to a minimum using lightweight symmetric encryption functions while they rely on unconstrained coordinating nodes in concert with a security manager service to manage authentication, key distribution, and security oversight.

Within this framework, I present an approach to securing the IoT by exploiting the capabilities of ICNs. Its contribution is an evaluation of the current state of ICN-IoT security architectures and their scalability and a framework with supporting protocols that support a range of security services for secure initialization of constrained devices into an ICN-IoT in a way that is efficient and scalable. In Chapters 5 & 6, I evaluate this work in terms of security, scalability, and efficiency.

3.2 Related Works to this Contribution

Previous work has established the characteristics of ICNs that make them well suited for use in the IoT. [9] established a mapping of IoT challenges to ICN features in terms of scalability, naming and addressing, mobility, security and privacy, heterogeneity and interoperability, data availability, and energy efficiency. The authors of [55] provide a similar overview of the IoT and its unique communication models, physical layer connectivity, and application requirements. They map these requirements to the capabilities provided by ICNs, then discuss the realization of the IoT through ICN in domain-specific use cases.

The authors of [10] performed real-world experiments demonstrating ICN usage for a small scale experimental IoT implementation. They demonstrated the feasibility of ICN with decreased control traffic and efficiency in energy and bandwidth constraints in a deployment across several rooms of a building. However, their implementation did not address security concerns. Several architectures have been proposed for custom IoT applications [56][57][58] and generic IoT as in [10]. These have limited focus to specific security imperatives such as data delivery, service discovery, or similar higher-level concerns instead of IoT device initialization, authentication, and security configuration.

Device initialization and onboarding in ICN-based IoT has also been given attention in work, such as [57]. However, the architectures rely on computationally intense security mechanisms based on asymmetric encryption used throughout the NDN stack. These mechanisms were evaluated in [59] in terms of energy and computation

time. They determined there was too high of a cost for resource-constrained devices, such as those used in many IoT use cases and that they would not scale well to large deployments. To address this, [60] introduced a design that uses symmetric cryptography based on AKEP2, enhanced with greater efficiency to work in an ICN.

However, in [60], the security model is limited to initial authentication and does not address secure forwarding and routing or any other security requirements. This requires that it be used in conjunction with separate frameworks to address other security concerns, which increases the overhead required in both processing and network bandwidth. In [61], the authors extend the work of [60] with a similar approach to authentication, but also propose a novel scheme to setup secure routing in conjunction with device authentication. Their framework was shown to be highly scalable with similar architectural structure to my work, but the protocol is restrictive in the network structure and limits the mobility of devices on the network. Further, it does not address additional security concerns and does not address a range of threat scenarios. More recent work in the NDN-Lite project [62] aims to provide an architecture and full library stack to support device authentication, bootstrapping, service discovery, etc. However, that architecture relies on asymmetric encryption, shown in [59] not to scale for resource-constrained devices, and targets a smaller scale Smart Home-like usage scenario where devices are managed locally [57].

3.3 Architecture Contribution

Several NDN/ICN-based IoT architectures have been proposed in the literature [63] [3]. However, none of them addresses the full range of security concerns from device initialization, network authentication, secure routing, and service discovery in a way that is scalable to the level of a smart city. I propose a framework and supporting protocols that address all of these concerns in a way that is secure, efficient, scalable, and compatible with constrained devices.

Similar to some of the prior proposals, I use a mesh network with a hierarchical network structure (Figure 6) that has been shown to support highly scalable applications for IoT [60] [61]. This structure allows use of coordinating nodes (CN) that are not resource-constrained. Such nodes can offload the processing and storage requirements for routing, authentication, and access control from the constrained IoT device nodes. Each CN acts effectively as a sink as described in other network protocol nomenclatures [15] and is critical for coordinating activities for its constituent constrained devices.

CNs have additional storage to cache NDN data packets from their enclave of connected devices. They also can act effectively as fog nodes, which are less resource-constrained devices located at the network edge capable of providing resources to other constrained devices located near them without having to leverage distant cloud-based systems [64]. The interconnected CNs form enclaves that connect to a wide area network or Internet through gateway nodes that are also assumed to be unconstrained. Enclaves can represent a smart home, smart building, or a geographic region of a city.

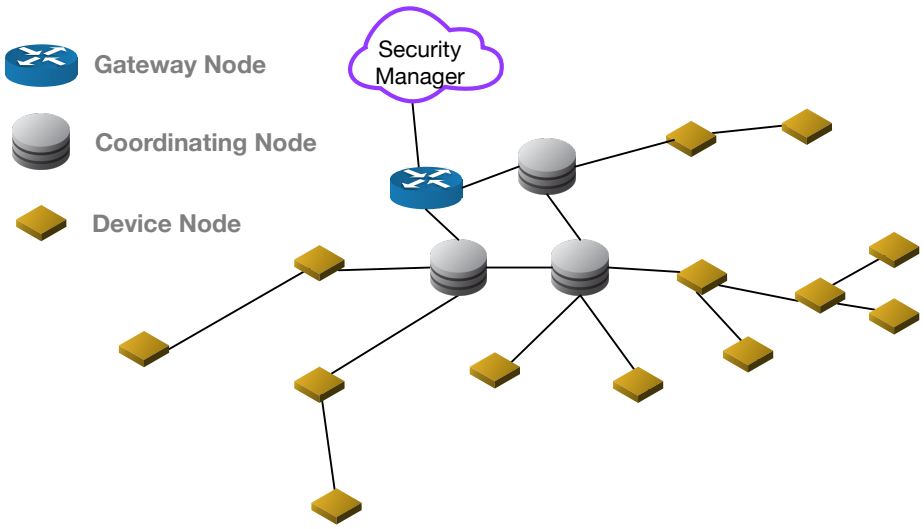

Figure 6. Network Architecture: ICN-IoT Enclave

The device nodes are required only to form a tree-like communications network, similar to that used in IPv6 with RPL, and have no expectation of extensive processing capabilities. Each device node first discovers a neighboring device and must authenticate the network itself and then itself to the network before continuing the initialization process. This is critical in preventing untrusted devices from launching denial of service attacks on the network (e.g., interest flooding or link exhaustion) and protecting privacy attacks. Lightweight symmetric encryption with pre-shared keys (PSK) is used rather than asymmetric encryption, in order to reduce the resource requirements for the device nodes. Security management is controlled by the controller nodes and a Security Manager (SM) service, which may run as a cloud service.

Because the CNs and SM are assumed to be unconstrained, it is assumed that CNs may encapsulate interests and data content from the enclave network with asymmetric cryptography when transmitting it to the SM. Similarly, they may authenticate and decrypt interests and data from the SM or other CNs before relaying them into their enclave network using symmetric encryption.

3.4 Smart City Use Case

Here I consider the proposed architecture in the context of a smart city. The IoT devices in the smart city are composed of many constrained device nodes that will be spread geographically throughout the city in varying densities depending on the applications. A smart city may include the below categories of devices and sensors:

1. Road traffic sensors that are used to monitor traffic to intelligently manage congestion and optimally update traffic light timing.
2. Smart parking sensors and devices to track and monitor parking availability throughout the city with capabilities for real time notifications for drivers to locate available traffic spots and by city managers/planners to identify regions of the city in need of additional parking resources.
3. Public transportation management using sensors and tracking devices to monitor usage, location, and congestion on buses, subways, and other public transportation modes. These may be used to optimize punctuality and availability of those services.

4. Street lighting sensors manage the public lighting resources to optimize security and sustainability by adjusting lighting schedules based on the detection of people or vehicles or in response to public security concerns.
5. Utility management allows for smart metering, smart billing solutions, and identification of consumption patterns with remote monitoring for gas, energy, and water usage.
6. Waste management allows sensors and monitors, e.g. fill levels of waste containers, to optimize waste collection to reduce fuel and energy consumption when containers are empty or not filled beyond some threshold. Additionally, it can identify locations that need additional waste containers.
7. Environmental sensors throughout a city include air quality monitoring, water quality monitoring to identify citizens in the city of potential hazards or dangerous events.
8. Weather monitoring sensors and devices to increase the accuracy of meteorological data and respond to changing conditions.

So in the context of a smart city based on the proposed architecture, these various sensors and IoT devices would be part of IoT network enclaves distributed throughout the city. Each enclave would have one or more coordinating nodes operating on non-overlapping radio channels to manage its constituent IoT devices. A single smart city could consist of hundred or thousands of these enclave networks to encompass all of these types of sensors and devices. All of the enclave networks are interconnected

through their gateway devices and the cloud security manager, so they can be securely managed and utilized to provide services for the city and its residents. All of this is done using Information-Centric Networking and Named-Data Networking.

3.5 Framework and Supporting Protocols

This section provides an overview of a scalable ICN-IoT framework, describes the relevant cryptographic mechanisms and protocols it relies on and provides a detailed description of how a device node joins the network, registers its services, and requests services to consume. The model assumes the device nodes are resource-constrained. However, they have the capability to perform some necessary symmetric encryption, such as AES, and hash functions as required to perform message authentication through the use of keyed-hash functions. It is assumed that these underlying functions and algorithms are not compromised.

3.5.1 Framework Overview

There are four stages to a device node securely joining the network and registering services (shown in Figure 7). In the first stage, the device node broadcasts a discovery request to identify a neighboring device that is already successfully joined. In stage 2, the new device node authenticates itself to the network. In stage 3, the secure forwarding paths are established in similarly as described in [61], but with service metadata also provided to the device node. In optional stage four, service discovery is

completed for services requested by the new device. These stages are described in detail in the following sections.

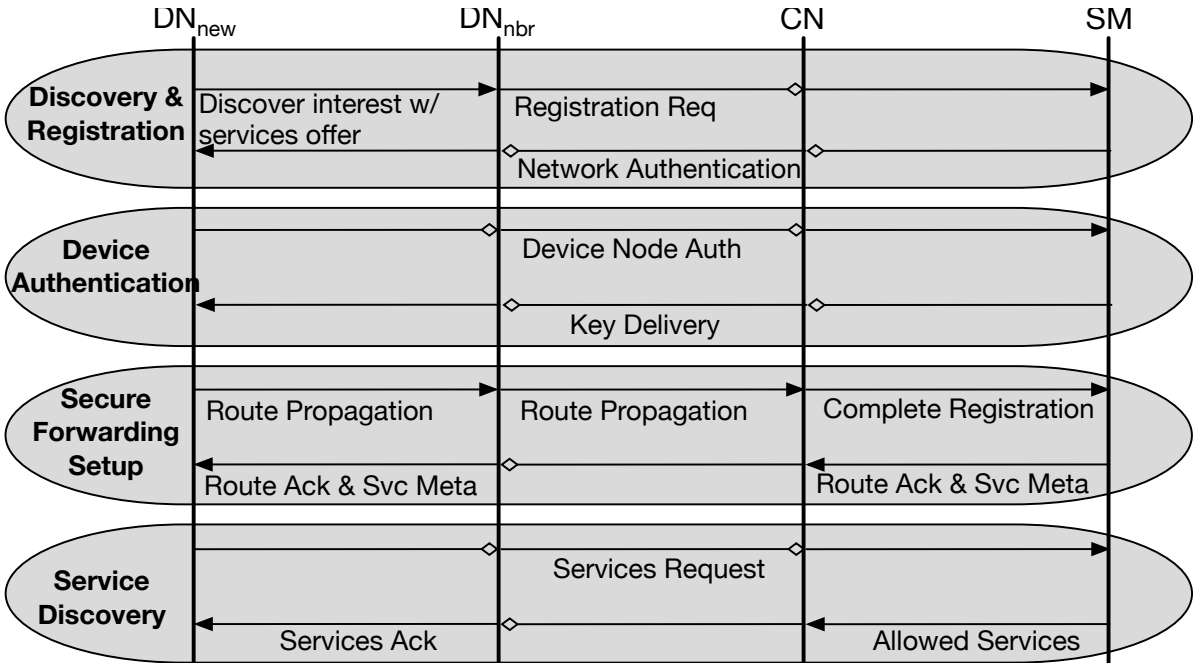

Figure 7. Stages of a device node joining a network

3.5.2 Encryption and Keys Overview

Our framework relies on symmetric encryption algorithms dependent on a tree of keys as shown in Figure 8. Central to this design are two permanent pieces of information that remain on each device node: an arbitrary identifier IDDN and a random pre-shared key PSKDN. This information may be embedded in the devices when they are

manufactured or there may be an ability to update them later. I assume a separate process exists to register these with the SM. As an example, this may be through a mobile application that could scan a QR code from the device node or another manual process. If they are changed later, there must be a process for updating them in the Security Manager, but these processes are beyond the scope of this work.

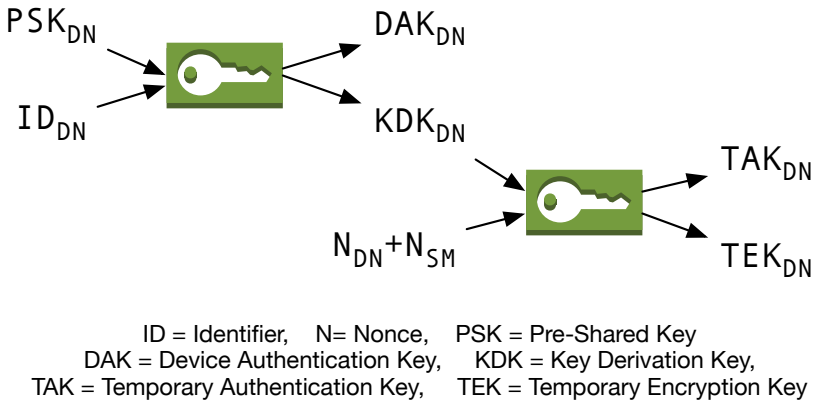

Figure 8. Password Derivation Process

My framework uses a key hierarchy and derivation process similar to that used in [60] and [61], which are based on the authenticated key exchange protocol (AKEP2) and the key extensible authentication protocol (EAP-PSK). However, rather than use the existing pre-shared key for operations directly, I use the two permanent pieces of information on the device nodes to derive two sets of additional symmetric keys. To accomplish this, I use a key derivation function that applies a pseudorandom function to

calculate the keys based on PBKDF2 [65]. Alternatively, it is possible to use an alternate derivation function such as scrypt [66]. The same derivation scheme must be used on both the device and the Security Manager. The parameters of the PMKDF2 function are shown below in Table 1. Password Derivation Function Parameters. The output from this function is 256 bits of keying material, split in two to derive two 128 bit keys.

Table 1. Password Derivation Function Parameters

Pseudorandom function	HMAC-SHA256
Password	PSK_{DN}
Salt	ID_{DN}
Iteration Count	1000
Octets Derived	32

As shown in Figure 8, the pre-shared key PSK_{DN} is used to generate two long-lived keys. DAK_{DN} is a device authentication key used to authenticate the device to the SM through an HMAC-SHA256 signature algorithm. The key derivation key KDK_{DN} is used with the nonces computed by both the device and the SM, then shared during the device discovery and registration process. They are used to generate two transient keys through the same function from which DAK_{DN} and KDK_{DN} were generated, except the salt used is the combination of nonces. In cryptography, a salt is a random sample of data that is used as an additional input in a one-way function that is used to generate hashes, passwords, or passphrases. Salts are useful for safeguarding passwords or keys in storage.

The transient keys produced are TAK_{DN} and TEK_{DN}. TAK_{DN} is used to authenticate data using the HMAC-SHA256 signature algorithm while TEK_{DN} is used to encrypt data using AES128-GCM and then encrypt the network forwarding key for the enclave network ($NFK_{ID_{CN}}$), which is sent to the device node after the registration process is completed and the device is authenticated.

It is important to note that the framework is extensible and can work with other lightweight cryptographic algorithms, such as those currently under consideration for the National Institute of Standards and Technologies (NIST) Call for Algorithms for Lightweight Cryptography, which currently has ten finalists under consideration [67].

The network forwarding key ($NFK_{ID_{CN}}$) is used later to establish secure forwarding paths as well as to authenticate new device registration requests. This process is described in detail in the next section. The long-lived keys are only used for device authentication and key derivation, while the transient keys are used for signing and encrypting application data. The utility of the transient keys is in that they can be recomputed if one is compromised by generating and sharing new nonces between the device and SM. A special interest can also be sent to the SM periodically to refresh the transient keys by exchanging new nonces without having to repeat the entire initialization process. In addition to these temporary keys, the SM may provide the DN with service keys which can be used to authenticate and encrypt interests and data for services the device node is permitted to provide.

3.5.3 Framework and Protocol Walkthrough

The framework relies on a set of protocol-based exchanges to onboard the devices into the ICN and provide security. In this section, I walk through the stages of the protocol from Figure 7. The parties involved in this process are shown in Figure 9. These include a new device node trying to join the network (DN_{new}) a neighboring node that is already a part of the network (DN_{nbr}) a coordinating node, CN, and the security manager, SM. The SM may exist as a cloud service in the internet. As previously discussed, the device nodes are constrained devices, such as wireless sensor nodes. It may be necessary for them to broadcast a beacon or alert message to wake up neighboring devices prior to beginning this process. There may be additional DNs on the path to the CN.

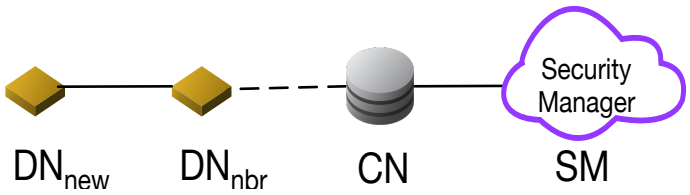

Figure 9. Parties involved in device integration

3.5.3.1 Stage 1: Discovery & Registration

In the first stage, a new device node that has never previously been a part of this enclave attempts to discover neighboring devices, authenticate the network it is attempting to connect to, and in the process, register the service capabilities it offers with the SM and the CN. This requires the cooperation of another device node that is already authenticated to the network. Figure 10 shows the details of the first interest exchanged in this stage.

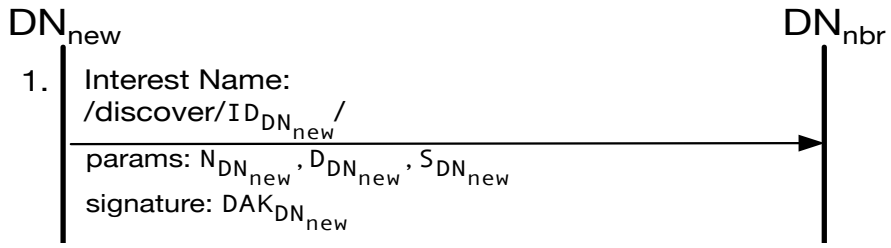

Figure 10. New device node sends an interest to discover the network

In the first step of the first stage, the new device DN_{new} broadcasts a *discover* interest to be received by nearby device nodes combined with its ID as /discover/ $ID_{DN\text{-}new}$/. Included with this request is a unique random nonce that it generates for each *discover* broadcast as $N_{DN_{new}}$, a hop distance from its coordinating node, which is initially -1 for a new device node, and an optional list of services that this device can provide as $S_{DN_{new}}$. A signature of the contents of the interest is computed from $DAK_{DN_{new}}$and included with the *discover* request. A neighboring node receiving the *discover* request

will not be able to verify the signature, but can encapsulate the request and pass it on to the CN and then to the SM.

When a neighbor node DN_{nbr} receives the request, it identifies that this request is for a new device by checking the distance provided and using its own distance to see if it's suitable to serve as a relay. If so, it will repackage the *discover* interest into a *register* interest and relaying it to the SM via its CN. This new interest is shown in step 2 of Figure 11. The original *discover* interest from the new device remains cached in DN_{nbr}. Its pending interest lifetime should be set relatively long to allow the process to complete.

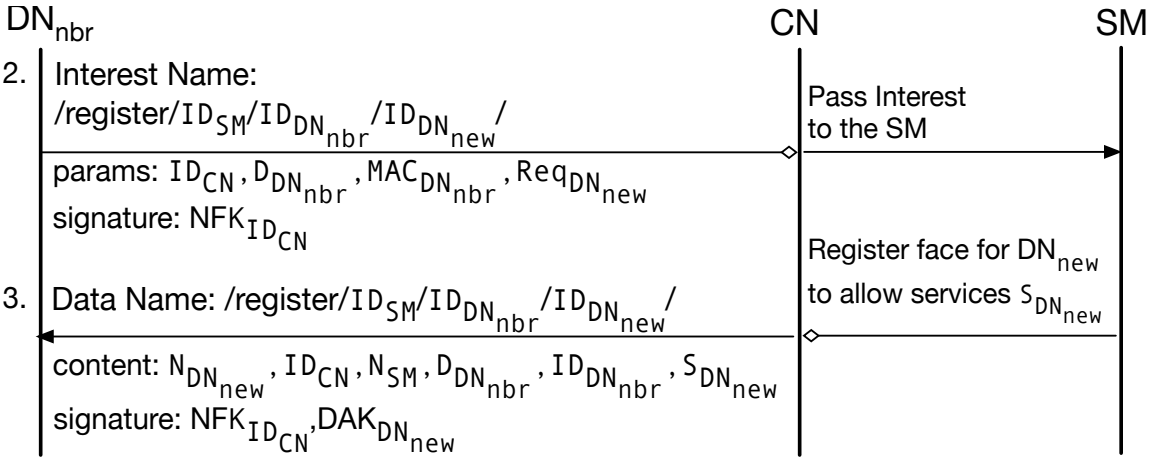

Figure 11. Neighbor sends a request to the SM to register a new device

The prefix for the *register* interest is simply /register followed by the ID of its SM and that of the new device as /register/ID_{SM}/$ID_{DN_{new}}$. The parameters included include:

- The ID of the coordinating node ID_{SM}
- The distance in hops between the neighbor node and the CN
- The ID of the neighboring node $D_{DN_{nbr}}$
- The original discover request with signature from DN_{new}

All of this signed by the $NFK_{ID_{CN}}$, which is the network forwarding key shared by all in the tree of authenticated and registered device nodes coordinated by this CN.

The signature in the *register* interest is validated before propagation by the CN, as well as by any additional DNs through which it might be relayed. Interests with invalid signatures can be discarded and eventually, their cached pending interests will expire. After receiving the register request, the SM will again validate the main signature and parse out the original discover request from DN_{new} to use $ID_{DN_{new}}$ to retrieve its corresponding pre-shared key already known to it. It will use this along with the device's $ID_{DN_{new}}$ to compute the new device node's $DAK_{DN_{new}}$ and $KDK_{DN_{new}}$ as described previously. The DAK is then used to verify the signature of the original request confirming it came from the specifically approved device. The SM will then review the list of service capabilities in $S_{DN_{new}}$ with those associated with the new device. If the list of advertised services correctly matches, it will compute its nonce and generate a data reply to the register interest of the same name, as shown in step 3 of Figure 11.

The data reply named for /register/ID_{SM}/$ID_{DN_{nbr}}$ /$ID_{DN_{new}}$ uses the same prefix combined with the ID of the SM, the neighbor's ID, and the ID of the new device node. The payload of the data includes the components of the original register request from DN_{nbr} except for the original discover message from the new device. The message also includes two signatures. The first is the $NFK_{ID_{CN}}$ used by the CN to validate that the data reply is from the SM. Once it has been validated, it will review the list of service capabilities and associate them with the appropriate face with the ID of the device node. The CN will not propagate data provided from a device ID for services that were not approved by the SM.

The second signature is used only on the combination of the $ID_{DN_{new}}$ in the data name combined with the content of the register response. This is because DN_{nbr} will need to create a new data reply to forward to the new device in response to the original discover interest still in its pending interest table, as shown in Figure 12. However, before doing this DN_{nbr}, as well as any other device nodes in the path, may verify the first signature using their copy of $NFK_{ID_{CN}}$ before forwarding.

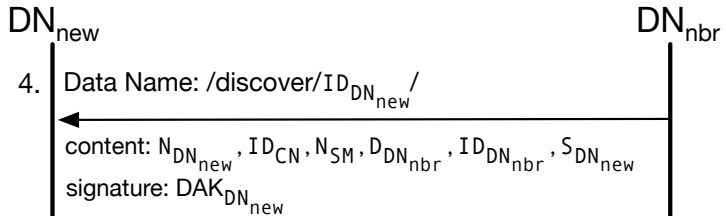

Figure 12. Neighbor DN responds to the discover request

In the final step of this stage, DN_{nbr} forwards the data reply named /discover/ $ID_{DN_{new}}$/ to the new device DN_{new}, which it can verify came from the correct SM by checking the signature created using its $DAK_{DN_{new}}$, and if verified correctly, it will have successfully authenticated the network and can proceed to the next stage. The new device node may repeat the discovery process iteratively to find a shorter path to its CN by sending a new *discover* interest with new nonce and hop distance.

DN_{new} may be in proximity to the CN initially, and in this case, it would process the *discover* interest and convert it to a *register* interest to pass to the SM, thus performing the work that would have been done by DN_{nbr} and proceed with the next stage.

3.5.3.2 Stage 2: Device Authentication

The purpose of this stage is to complete the authentication of the new DN to the SM and the enclave network. At the end of this stage, the new DN will also receive the network forwarding key shared by all devices within this enclave network. The steps involved in this stage are shown in Figure 13 below.

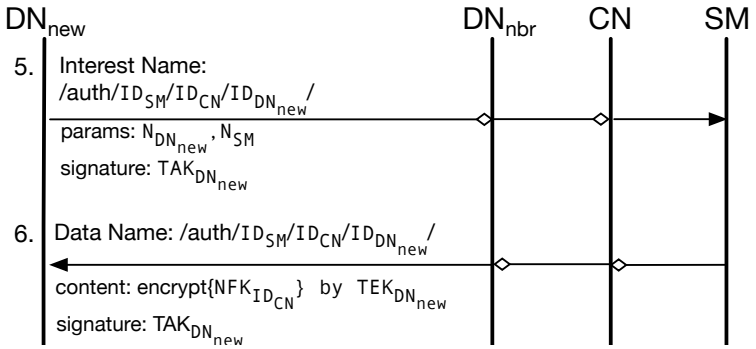

Figure 13. New DN sends interest to authenticate to the network

The SM was already able to authenticate the new DN by verifying the signature of the original request signed with $DAK_{DN_{new}}$ in step 1 and encapsulated in the *register* interest from the neighboring DN. The data content sent in response to this from the SM includes a nonce, N_{SM}, generated by the SM as a challenge to confirm authentication and to thwart potential replay attacks of the original discover request. The new DN must use this nonce, as discussed previously, to produce transient keys to match those generated by the SM.

First, the new DN will create an *auth* interest directed to the SM via the enclave network as shown in step 5. This will be propagated up the existing forwarding path and cached throughout the corresponding tree of devices until it reaches the SM. The parameters of this interest are a repeat of the nonces of both the new device and the SM for this device, which should match those already known to the SM for this device. The new DN will sign the *auth* interest using $TAK_{DN_{new}}$, its temporary authentication key.

When the SM receives the *auth* interest, it will first verify the nonces match the ones known to have been last used for this DN. It will then use its copy of $TAK_{DN_{new}}$ to verify the signature. If successful, it will provide a content response that includes $NFK_{ID_{CN}}$, the network forwarding key. To protect this key in transit, it is encrypted with $TEK_{DN_{new}}$ while the content is signed with $TAK_{DN_{new}}$, both already known by the new DN. This will allow the new device to proceed to the next stage and eventually allow it to assist other new DNs in joining the network.

3.5.3.3 Stage 3: Secure Forwarding Setup

The new DN now has enough information to send interests into the enclave network; however, the enclave network does not have enough information to forward interests to the new device. The tree of devices to which the new device is connected also does not know what services they should trust from the new device. In this stage, the DN establishes a secure forwarding path through the tree of devices between it and its CN by sending a *route* interest (seen in Figure 14).

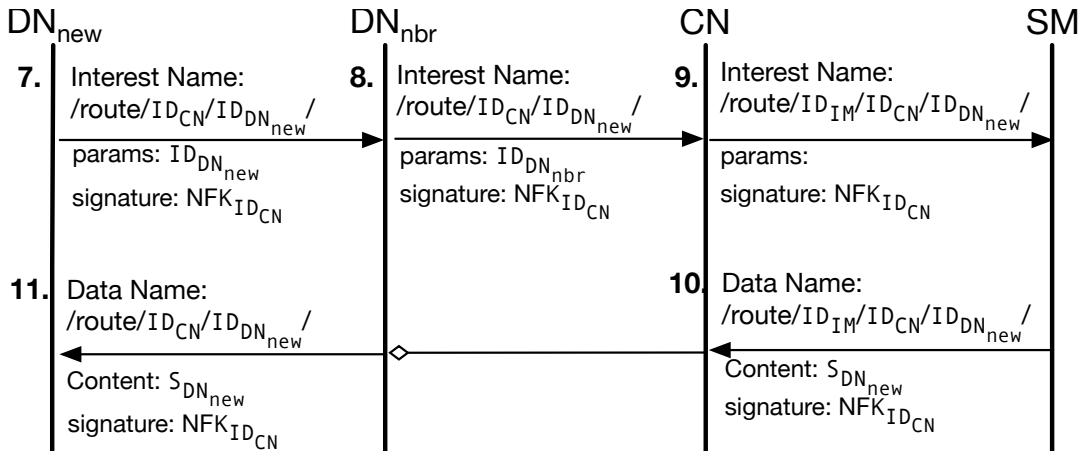

Figure 14. New DN defines a forwarding path to its CN and service availability

In my framework, the CN maintains a custom forwarding information base (FIB) of each of its constituent DNs, the service interest they're authorized to provide, and the next-hop MAC address to reach them. Each DN also maintains this information in its forwarding table for the DNs in its subtree. The DNs employ a forwarding strategy that consults the FIB to make forwarding decisions directed to DNs for specific services. To register itself on its parent node's FIB, each DN will send a *route* interest that includes its ID and that of the ID of its CN in the interest name. It will also include its MAC address as a parameter and sign the request with the $NFK_{ID_{CN}}$ obtained in the previous stage, as shown in step 7. Then as each parent DN receives this request, it will verify the signature, update its FIB, and generate a new *route* interest replacing the MAC address parameter with its own, passing it to its parent DN or eventually to the CN as shown in step 8.

When the CN receives the route request, it will update its FIB and generate a new interest for the SM. The SM will perform its verification and then generate a content

reply for the CN. The content reply includes a list of service prefixes that DN_{new} is authorized to provide. It is assumed that this information is known to the SM already or entered out of band similar to ID_{DN} and PSK_{DN}. The CN will then generate a content reply to fulfill the pending interests of the original *route* interest and forward it according to its FIB. As each DN along the subtree receives the reply it will validate it and update its FIB with the services. Each DN will only propagate interests to or content from a DN with service prefixes that match its FIB. At this point, DN_{new} is part of the enclave network and capable of registering its own new neighboring devices.

3.5.3.5 Stage 4: Service Discovery

At this point, the new device node is bootstrapped into the network, can communicate with the enclave network, and other devices can communicate with it. The final stage shown in Figure 15 is optional and determined if the device needs to request access to a service. The service may be provided by another DN, the CN, or from another enclave network.

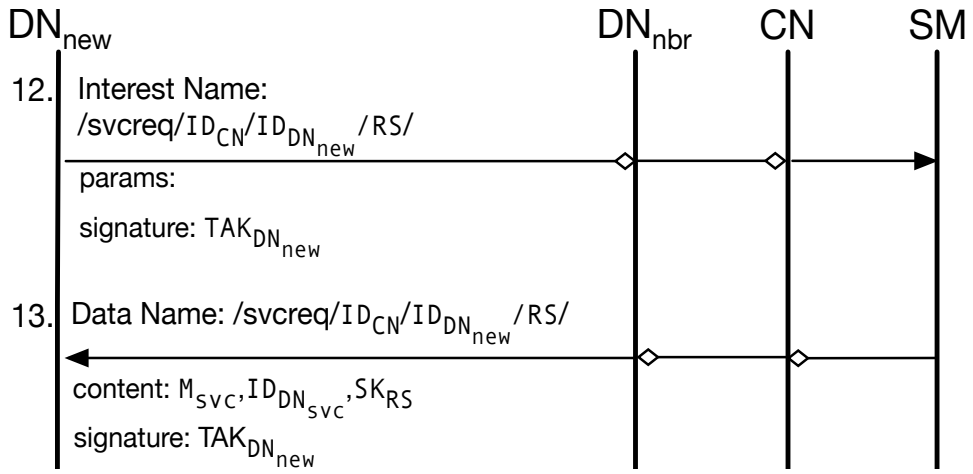

Figure 15. New device node can optionally request access to service.

To request access to a service, the new device node will send a *svcreq* interest directed to its CN, which includes the name, the CN's ID, its ID, and the requested service, RS. It will sign this with $TAK_{DN_{new}}$ to authenticate its request. The data content of the reply from the SM will include metadata required for the new device to use the service correctly. It may also include an explicit ID for a provider to request the service and a service key to authenticate the request and decrypt the data content for the service. When the CN sees the data reply, it will update its forwarding information base to allow it to propagate interests for the named service from the new device. After this stage is complete, the new device is fully registered as part of the enclave network.

3.6 Summary

In this chapter, I have presented a framework and supporting protocols that extend prior works to address the authentication, registration, secure forwarding, and service authorization and discovery of constrained devices into an ICN based IoT in a way that is highly scalable. I leverage a mesh network with a hierarchical structure to enhance scalability. The device nodes participating in our architecture are assumed to be constrained, so cryptographic operations are kept to a minimum using lightweight symmetric encryption functions while they rely on unconstrained coordinating nodes in concert with a security manager service to manage authentication, key distribution. and off load computation from the devices. This work provides a novel way for constrained devices to securely and fully integrate into an ICN based IoT efficiently. This includes not just authentication and secure routing, but also service authorization and discovery, using an architecture that can scale for use in a smart campus or city.

4. ENHANCING THE ICNOT WITH TRUST-BASED ACCESS CONTROL

This chapter expands on my prior work on architecture and supporting protocols to efficiently integrate constrained devices into an Information-Centric Network-based Internet of Things (ICNoT) in a way that is both highly secure and scalable. In this work, I add new capabilities for addressing additional threats and integrating trust-based behavioral observations and attribute-based access control by leveraging the capabilities of less constrained coordinating nodes at the network edge close to IoT devices. In this edge computing model, these coordinating devices have better insight into the behavior of devices and access to a trusted overall security management cloud service. I leverage two modules, the security manager (SM) and trust manager (TM). The former provides for data confidentiality, integrity, authentication, and authorization, while the latter analyzes the nodes' behavior using a trust model factoring in a set of service and network communication attributes. The trust model allows trust to be integrated into the SM's access control policies, allowing access to resources to be restricted to trusted nodes.

4.1 Introduction and Overview

IoT has revolutionized many applications and industries with the exponential growth of heterogeneous interconnected devices that generate an ever-increasing amount of data supporting real-time processing into information and capabilities for sharing it instantly across advanced communications networks. This introduces new challenges in

processing the large volume of data and ensuring security, privacy, and trust over the devices and data.

ICN has emerged as a promising internet architecture that provides inherent benefits to the IoT, such as named data, in-network caching, mobility, and data-driven security rather than connection-oriented security. ICNs expand the network architecture from a host-centric one where communication is between endpoints to a data-centric one where named data is requested from the network itself. The many characteristics of ICNs that make them well suited for the IoT have been demonstrated in several works [10][68][9].

While the inherent properties of ICNs make them well suited for the IoT, there remain many challenges, especially concerning security, privacy, and trust. The large number of heterogeneous devices interacting includes many constrained devices with limited resources in terms of battery or energy, processing capacity, memory, and storage capacity, making them more susceptible to attackers. The nature of their interconnections and wireless communications, such as between sensors, switches, and actuators, make it possible to compromise them by various attacks, such as interception, denial of service (DoS), and various man-in-the-middle attacks. The goal of these attacks is to breach the basic security objectives of the system: confidentiality, integrity, availability, authentication, and access control.

In Chapter 3, I introduced a secure architecture and supporting protocols for efficiently integrating constrained devices into a pure ICN-based IoT. It includes mechanisms for authenticating the network to the device, the device to the network,

establishing secure forwarding paths, and registering a service for discovery into ICN-IoT enclaves at the Internet edge. These enclaves can be interconnected through cloud services to produce large scalable IoT networks.

In this chapter, I propose an addition to this architecture that supports capabilities for attribute-based access control using trust-based behavior observations of device nodes on the network. While the prior work introduced security mechanisms sufficient to protect against many external attacks, in this work I expand security against internal attackers by establishing mechanisms to establish and monitor the trust of devices based on their behavior and quality of service attributes over time. I also show how to integrate this capability to make encryption-based access control decisions based on this trust.

4.1.3 Organization

The chapter is organized as such: Section 4.2 provides for the reader general info on ICNs, then an overview of the prior chapter relevant to this work. Section 4.3 provides information on related work in this area and how it complements or differentiates from our contribution. Section 4.4 presents the proposed additions to the architecture and defines the new security and trust manager components. It also presents the trust attributes, parameters, defining equations and supporting algorithms, and related discussion. Section 4.5 provides a concluding overview of what was presented and identifies future work.

4.2 Background and Relation to Previous Chapter

To provide background, I begin by defining some of the basic components of an Information-Centric Network. Some of this is discussed previously in Chapters 1 & 2. While these aren't necessary to understand the contribution in general, they are useful for context. As stated, ICN is a rethinking of network communications from Internet Protocol (IP)-based to information-based. IP-based communication begins by first connecting to an IP address at the network level and then works the way up to making application-level requests such as a web search. Based on the results, the user may click a link to visit the website of interest, which involves a new connection to a new IP address resolved from the Domain Name System (DNS) address. In an ICN, this initial query would be made directly to the network itself by creating an *interest* packet. This would be forwarded through the network based on forwarding rules. At each forwarding node, a local cache is checked to see if the data requested is already present. If it is, it can immediately be returned as a *data* reply. If not, it is registered in a pending interest table (PIT) along with the face (ICN nomenclature for interface) on which it was received, then forwarded to the next appropriate face. When the data reply is eventually received, the face will first cache the data so that any future requests can receive an instant response. Then, it will check its PIT to forward the data reply to the appropriate face to make its way to the original requestor. In this simplified example, requests for data that is popular will get much better performance and reduce constraints on bandwidth by having the data already cached in the network.

In Chapter 3, I defined a scalable and efficient architecture and supporting protocols for integrating a constrained device into an Information-Centric Network of Things. This section will summarize the relevant aspects of this architecture and its stages. Figure 6 shows the mesh network with a hierarchical structure as the basis for the architecture. This architecture includes coordinator nodes (CN) that are *not* resource-constrained and constituent device nodes (DN) that are assumed to be constrained. DNs would include a range of devices such as temperature sensors, lighting controls, door locks, various actuators, cameras, and other devices. The CNs can perform complex processing for routing, authentication, access control and employ complex cryptography and machine learning. They have sufficient storage to cache data from their constituent devices and store observational data regarding the quality-of-service parameters for them. The CNs in effect act as a sink, as described in some other network protocol nomenclatures [15], and are critical for coordinating and orchestrating the constituent DNs. The CNs are assumed have reliable and secure internet connections via gateway node (GN) devices to a security manager (SM) that exists as a cloud service. The GW, CN, and connected DNs represent enclave networks that may be interconnected with many other enclaves to constitute a smart campus or smart city.

Refer back to Figure 7 in Chapter 3 for the stages in the architecture protocols in terms of the interest and data replies for a new IoT device node DN_{new} joining the ICN-IoT enclave network by first broadcasting a *discover* interest to identify a neighboring node and authenticating the network based on the reply. Given multiple replies, it can select a neighbor based on proximity, response time, or distance to the CN.

It then authenticates itself to the network through the CN. It next establishes a secure forwarding path to its CN and optionally requests a service.

The process relies on lightweight symmetric encryption and signature verification based on keys generated through a key derivation process using information added to the SM when the device is first added to the enclave. When complete, the new device has a set of permanent and transient keys that can be used to authenticate or encrypt data on the network, in addition to a network forwarding key, NFK, that can be used to authenticate other devices on the enclave network [69]. The NFK can be regenerated periodically, whereupon device nodes can rediscover to find improved paths to the CN.

The purpose of this chapter is to enhance that capability with additional security capabilities and create a more complete security architecture with robust capabilities for managing the whole process of device registration, authentication, network authentication, route management, and access control, including managing trust of the devices over time.

4.3 Related Works to this Chapter

Although there is considerable work in the rapidly evolving area of IoT, including in security, trust, and privacy, there remain many open research areas, especially concerning ICN-based IoTs. I discussed several related works in Chapters 1 & 2. In this section, we explore related research in both IoT and ICNs related to trust. Some of the advances in IoT security, trust, and privacy do not apply to ICN-based IoTs, while others have core capabilities that can be adapted to suit them.

[70] presents a secure surveillance framework for the IoT based on probabilistic image encryption using a mechanism of video stigmatization to extract frames and encryption of the images to prevent data modification attacks. A context-aware multifaceted trust framework for determining the trustworthiness of cloud service providers is presented in [71]. The trust level in a cloud service is calculated by factoring in both user experience and service characteristics and using fuzzy simple additive weighting. Additionally, [71] derives a service level agreement-based trust by using an analytic hierarchy process [72]. A hardware-agnostic security framework is proposed for fog-based IoT networks in [73]. It uses a set of detection, collection, and management tools and machine learning to identify and mitigate vulnerabilities present in a variety of IoT protocols, such as Z-Wave and LoRaWAN (a standard for Low Power, Wide Area Networking protocol). A graph modeling technique is used in [74] to identify the relationship between vulnerabilities in industrial IoT systems. The security concerns are constructed using graph-theoretic problems and propose risk mitigation techniques to remove attack paths with elevated risk and low hop-length. In [75], the study presents a key agreement framework that uses a mobile-sink strategy with extended user authentication to cloud service applications. It utilizes bilinear pairing and elliptic curve cryptography (ECC).

A trust model for social sensor networks that uses multisource feedback and a fog computing model is presented in [76]. The model includes sensor nodes that provide feedback about other nodes for each interaction. The trust factor for each node is calculated by aggregating feedback from multiple nodes. [77] presents a trust model for

evaluating the trustworthiness of a cloud computing service through the combination of subjective and objective evidence. [78] addresses security in the Internet of Vehicles (IoV) through a technique using digital certificates, prioritization rules, and reputation policies that use trust to detect hijacked vehicles. These trust mechanisms are generally optimized for IP-based communication.

[79] describes the qualities of Named Data Networking (NDN), another name for and popular architecture and implementation of an ICN, that make NDN beneficial to the IoT networks in their use of data as the primary element that can be redistributed and cached in the network. However, not evaluated in this work are routing threats that might occur during communications. A novel scheme that extends the NDN security capabilities to support trust schemas is presented in [80]. Their proposal uses a publish-subscribe to cryptographically and structurally validate a subscriber's incoming publications using an application trust schema. It is optimized for home and business IoTs with the assumption that most devices are not constrained.

4.3.1 Attribute-Based Encryption

My scheme for access control employs attribute-based encryption (ABE), which is a form of identity-based encryption (IBE) first introduced in [81] and [82]. IBE allows users to encrypt and decrypt messages based on a set of attributes and access structures. The scheme I employ is similar to Cyphertext-policy attribute-based encryption (CP-ABE), which is a type of ABE scheme where the key for decryption is associated with a set of attributes for the user or node it is assigned to. The encryptor defines the structure

for access to protect the data so that only a user whose attributes satisfy the access structure can perform decryption on the message. This makes it well suited for use in access-control mechanisms. Initial CP-ABE schemes relied on complex, expressive access structures with large decryption keys, making them ill-suited for use in resource-constrained systems. However, newer schemes [83] have been designed to address this using fixed-size keys optimized for constrained devices. Further work in [84] has optimized them further through elliptic curve cryptography (ECC), which has shown to be the most efficient public-key alternative for supporting security services in constrained environments, such as the IoT [85]. [86] provides a comprehensive survey of access control mechanisms available in NDN, which is a popular implementation of an ICN. They provide a useful taxonomy of access control types, such as encryption-based or encryption-independent. although they do not cover trust-based access control mechanisms. They review several attribute-based access control schemes applicable to our work, though they do not involve a trust calculation in determining access.

4.4 Proposed Security and Trust Enhancements Scheme

This section describes our trust management system as integrated into my Chapter 3, the system architecture, threat model, security manager, and trust manager.

4.4.1 Enhanced System Architecture

Figure 16 shows the integration of the Local Security Manager (LSM) and Trust Manager into the coordinating node (CN). I now distinguish between the cloud security

manager (CSM) and LSM. As shown, there are three layers consisting of the IoT device nodes, coordinating nodes, and the cloud security manager. As with our previous work, the device nodes consist of IoT devices such as sensors, thermostats, lighting etc, that are presumed to be resource constrained. The coordinating nodes are assumed not to be constrained. The CNs are responsible for monitoring device node (DN) behavior, trust computation and management, and identity and access management. They also provide more intensive encryption computations and can communicate using standard non-device node (NDN) communications with asymmetric signatures and encryption capabilities with the cloud security manager (CSM). The CSM exists as a consolidated computing platform consisting of remote computing and storage platforms integrated to provide various services for managing aspects of multiple distributed enclave networks of coordinating nodes (CNs) and their connected DNs.

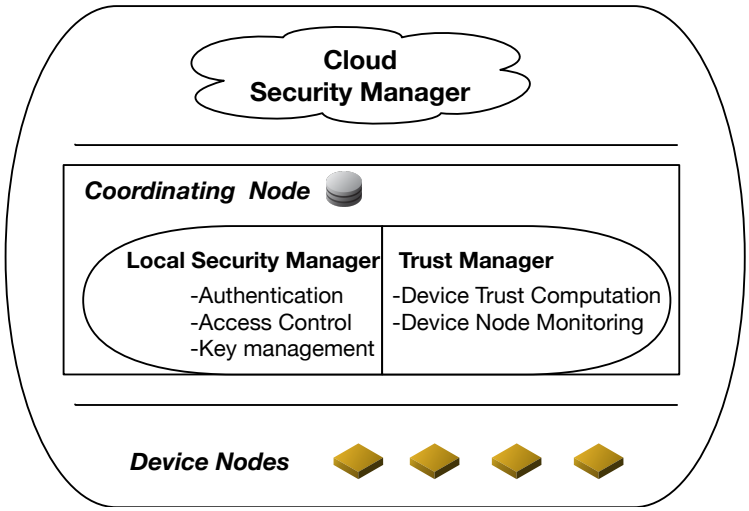

Figure 16. Trust and Security Integration

4.4.2 Enhanced Threat Model and Assumptions

The CNs must compute the trust of their constituent device nodes by monitoring their behavior. The CNs already track which DNs are associated with what face and can use this data combined with quality of service (QoS) observations to make decisions about trust. The threat model associates each node with a level of trust: trusted, semi-trusted, and untrusted. In the current model, I consider the CSM and CNs as trusted as their computational abilities provide for more robust security mechanisms. DNs are initially semi-trusted, but that status can change to untrusted based on trust calculations related to their behavior. Future expansion of this work intends to also consider the coordinating nodes as semi-trusted and provide for additional trust behavioral tracking of the CNs.

Some security threats include interception of unencrypted interest parameters, and data packets allow an attacker to break confidentiality and potentially integrity if they also lack signatures. Several malicious DNs could collaborate to perform denial-of-service (DoS) or distributed denial-of-service (DDoS) attacks. Less constrained devices could attempt to masquerade as compromised DN to also engage in these attacks. A DN device could perform sinkhole type attacks by not propagating interests and/or data either entirely as a black hole attack or selectively as in a grey hole attack [87]. Compromised nodes could attempt to corrupt the trust scheme by reporting false parameters. This necessitates the importance of maintaining continuous evaluation of the trust based on behavioral observations over time.

4.4.3 Security Management

This section describes the local security manager (LSM) within the coordinating node and the enhanced capabilities added in this work. Table 2 lists a set of symbols and their descriptions used in the remainder of this paper. The objectives for the security of the system are to provide confidentiality, integrity, authentication, and authorization, which are managed up front by the local security manager in coordination with the cloud security manager.

Table 2. Symbols and Descriptions

Symbol	Description
U	Universe of attributes
A	Device node attribute Set
P	Access policy
MPK/SPK	Master Public / Master Secret Keys
k_u	Device secret
C	Ciphertext
M	Interest parameters or data payload
w	Time window for trust computation
t	Time instance for trust computation
T_{DN}	Device trust for DN computed by CN
α	Weight for T_{DN}
$c(i)$	Parameters from the ith DN for the CN
σ	Std deviation of T_{DN} over time window w

Chapter 3 established a protocol for registering a device to the ICN-IoT that involved a unique identifier assigned to each device node as called for in [69]. In this system, it is important that each device node must have a unique identifier that is trusted. I extend this process in this work to include a set of attributes for each device node to be

used during registration to then be used in an attribute-based encryption function that generates secret keys that are then used to encrypt data and verify signatures.

Access control in the system is used to determine which device nodes (or other systems or users) can access other devices, resources, and services. The CNs can define and enforce access policies for all nodes in the system. If and only if a DN meets an access policy, can it perform an access operation on another DN or service, such as read or write on it. Table 3 lists a set of access operations and their trust attributes. In this enhancement to the prior work, I rely on an refined CP-ABE scheme previously mentioned and described in [84] that utilizes ECC and is implemented to provide robust authentication and access control in constrained systems like the ICN-IoT. This system improves upon the encryption system used in our previous work. This ECC-CPABE system comprises four algorithms:

Setup: This algorithm is passed a security parameter ***p*** and the universe of attributes $\boldsymbol{U}=\{\mathbf{A_1},\mathbf{A_2}\ldots,\mathbf{A_n}\}$ as inputs, and it outputs the key pair *MPK/MSK*

Encrypt: Inputs an access policy *P*, the *MPK*, and a plaintext *M*. The encryption algorithm *E*[*P,M*] then outputs a ciphertext *C*. This is used to provide data confidentiality for communications by encrypting messages between DNs and CNs.

KeyGen: Inputs of this algorithm are an attribute set *A*, *MPK,* and *MSK.* The key generation algorithm then outputs a user secret key to be used for decryption k_u that corresponds to *A*. These secret keys are used to ensure all device nodes are authenticated and legitimate.

Decrypt: This algorithm takes the ciphertext C produced with an access policy P, MPK, and k_u corresponding to attribute set A as inputs, and outputs the original plaintext M or otherwise outputs null ($\perp$) using decryption algorithm $D[C,P,k_u,A]$. If $P \subseteq A$, then $D[C,P,k_u,A]$ should always output the original plaintext M. This process enforces strong Table 3 authentication and authorization. Additionally, a hash function is used with this to provide interest payload and data integrity verification.

I integrate the trust attributes of the nodes into the ABE access control policies to be used for identity and access management. Table 3 lists several attributes, but there could be more or fewer depending on the environment. My current implementation primarily involves the first three: deny, read, or write concerning the device nodes. For this work, the coordinator nodes are considered trusted devices. However, it is possible the cloud security manager could conduct behavioral monitoring for a set of QoS attributes for the coordinator nodes as well. As trust increases or decreases, the level of trust assigned to a node results in privilege escalation or de-escalation. This is combined with a mandatory access control policy set in the LSM or CSM that can prevent elevated access for a node identity even with elevated trust.

As an example of the architecture, consider the universe of attributes U, device node attribute set A and the access policy P. We can represent A and P by a string of bits. For example, if $U=\{A_1,A_2,A_3,A_4,A_5,A_6\}$ and a device node has attribute set $A=\{A_1,A_2,A_4\}$, then a binary representation of the bits would be A=110100.

Consulting Table 3, we can consider an example representation of access rights that may be part of the system. The actual representation would vary based on the

on the actual devices and the nature of the enclave IoT network deployment and services that are offered. We can use the representation of the access rights for embedding trust in *A* and *P*. The binary representation of trust matches a level of access and is added to the device node's access policy and attribute set. If we consider a device node with a current Trust computation of 0.41, represented by attribute A_6, a device with 'write' access rights with binary representation 010 the bit string form would be *A*=11010010. If the access policy *P* is defined over $\{A_1,A_2,A_4,A_6\}$ and A_6 corresponds to "read" access or 001, then its access policy bit representation is *P*=11010001. As described in the "Decrypt" algorithm above, a device node with attribute set *A* meets the access policy *P.* That is, if $P \subseteq A$, then the device access rights are granted. The access rights for a device node are determined by the trust value that must be greater than or equal to the access policy or access denied.

Table 3. Attributes of ICN-IoT Nodes and Trust Attributes

ID	Access	Trust Score	Representation
01	Denied	<=0.3	000
02	Read	0.3>0.4	001
03	Write	0.4>0.5	010
04	Delete	0.5>0.6	011
05	Execute	0.6>0.7	100
06	Modify Config	0.7>0.8	101
07	Special Perm	0.8>0.9	110
08	All	0.9>1.0	111

4.4.4 Trust Manager

This section provides an overview of the trust manager component in the coordinator nodes. As previously described, the trust calculations from this component are used by the local security manager to make access control decisions cryptographically. My proposal is that this is a modular component with general characteristics that would be modified depending on the deployment environment and use cases. Here I generally describe the subcomponents of the TM and a generic set of variables to monitor on the device nodes for illustrative purposes.

The TM includes a *QoS monitor,* which tracks all ICN *interests* and *data* packets moving thru and associated with device nodes registered to the network and associated with one of its faces. A generic set of attributes or quality of service characteristics that might be monitored for a device node is presented in Table 4. In this example set, I include the throughput, which is the amount of data in interests and data packets transmitted by the device node. The bandwidth or the maximum data volume capacity of the device node. The energy consumption is considered where it can be obtained directly or calculated based on the number of transmissions from the device nodes. The trust manager monitors the number of interests and data requests sent or fulfilled for each device. The QoS characteristics includes the average time spent in the pending interest table and cache, respectively. In addition, the TM may collect statistical info on resources spent servicing requests from the DNs. It can also utilize information determined about the subtrees formed and distance in hop count to a device node in the hierarchical network and from other coordinator nodes that make up the enclave IoT network at the

Internet Edge. In addition, there may be another set of application level QoS characteristics that are tracked and included in the behavioral monitoring of the device nodes over time.

The trust of a device node is computed based on the QoS properties monitored by the CNs. That trust is computed for a time instance *t* over a time window *w* based on those properties. The time window may depend on different deployment scenarios and use cases e.g., minutes, hours, or days. Therefore the trust computation for individual device nodes may change over time as their behavior and interactions with other device nodes and coordinator nodes are monitored.

Table 4. QoS Trust Properties to Monitor

Trust Properties for Coordinator and Device Nodes
Throughput
Bandwidth
Energy Consumption
Number of interest requests per DN
Number of data replies to DN
Avg time interests remain in the PIT
Avg time data remains in the cache
Device location in subtree topology
Distance to node in hop count

Figure 17 shows a calculation to generate an instant trust score $\mathrm{T}_t(d_i)$ of an IoT device node d_i at time instance t. The calculation is based on the set of QoS attributes a over the time instance of interest. It is expected that this calculation would involve the

use of a machine learning algorithm, such as a random forest regression model, that has been shown to be optimal for use in making predictions based on various network performance characteristics in the IoT [88].

$$T_{DN} = \frac{\sum_{i=1}^{a} \mathrm{T}_t(d_i)}{a}$$

Figure 17. Trust Score Calculation

The TM uses a *Trust Matrix* to store the computed trust calculations for each device node. Any device node, coordinator node, or the cloud security manager can query to obtain the current trust calculation for a device node by sending a special *gettrust* interest for the enclave network and device node ID. When processing specific service interests, the coordinator node will consult the TM to determine the device node with the highest trust to rely on to facilitate that service interest. A trust credibility assessment model can be used to mitigate attacks and to ensure an accurate trust calculation. The trust credibility model modifies the trust of device nodes when they could be compromised. Trust credibility evaluation is applied in all calculations and if there is a significant variation, it can be adjusted. Trust is expected to increase or decrease as the enclave network operates.

The model evaluates the change in trust (T) over a time instance [t_0,t] and later recalculates the trust for a recent time instant t using the left equation in Figure 18. The standard deviation σ in T over a time window n informs about the spread of the potential values of trust and is computed by the right equation, in which μ is the mean of trust T at a specific point in time t. The standard deviation should be evaluated every time a new trust score is calculated. If the trust T in a very recent time instance t is less than the previous time t_0 and the difference exceeds the standard deviation, then T at a point in time t is increased. If not, then it is decreased.

$$\mathrm{T}_t = \mathrm{T}_{t_0} \pm \sigma \mathrm{T}_t \qquad \sigma = \sqrt{\frac{\Sigma(\mathrm{T}-\mu)^2}{n}}$$

Figure 18. Trust credibility evaluation model

4.5 Summary

This chapter has described enhancements to my work in Chapter 3 and supporting protocols for fully integrating constrained devices into an Information-Centric Network of Things in a way that is secure and efficient, and suitable to scale to smart campuses or cities. The previous chapter supports authenticating the network to the new device, the device to the network, establishing secure forwarding paths, and registering a service securely. I introduced security mechanisms sufficient to protect against many external

attacks, I expand security against internal attackers by establishing mechanisms to establish and monitor the trust of devices based on their behavior and quality of service attributes over time. I also showed how to integrate this capability to make encryption-based access control decisions based on this trust. I discuss a set of QoS attributes related to ICN and NDNs that may be used and provide an example of how they may be used to calculate a trust score for a device node based on them over a window of time. Finally, I introduce example equations for calculating a trust score for a time instance using a trust credibility model.

5. SECURITY EVALUATION

5.1 Informal Threat-Based Security Evaluation

This section will provide some discussion of various attack scenarios relative to this framework and protocols. The types of threats and attacks on IoT devices and IoT environments are studied extensively in the literature. I use a set of attack scenarios similar to those presented in [89].

1. External attacker passive monitoring network traffic

There is a little that can be done to prevent attempts to intercept signals for traffic analysis in a distributed wireless network. However, the use of encryption using both pre-shared and derived keys protects the confidentially of important data and key materials as they are distributed.

2. External attacker attempting to fraudulently join the network

A malicious device node could attempt to join the enclave network by sending a *discover* interest to neighboring device nodes. The neighboring node would pass this as a register interest through the tree to the CN, which in turn would pass it to the SM. The SM would not have a matching ID and PSK for the malicious device; and therefore, would not be able to derive a matching $DAK_{DN_{new}}$ to validate the signature from the interest. The SM would respond to the neighbor device with a data response prompting

them to remove the *discover* from their PIT ending the process and preventing the malicious device from joining the enclave.

3. External attacker attempting to masquerade as a legitimate node

In this scenario, a malicious device attempts to masquerade fraudulently as a neighboring device that is already a part of the enclave network to a new device node. Essentially it is trying to get a trusted device to join a malicious network. This would fail because the malicious device node would not have the pre-shared key of the new device and could derive the key to sign the data response to move to the next stage.

4. Inside attacker provides false data

An inside attacker is a compromised device node that was registered and onboarded into the enclave network. An inside attacker would be able to provide false data for the services for which it has been authorized. This could be difficult to detect but could be mitigated by the CNs and SM monitoring the data being provided by the device and comparing to predefined parameters and triggering an alert for human intervention.

5. Request data from services or devices without authorization

An insider attacker could send interests to a service on a specific device node for which it has not been authorized. If the service requires authentication via service key, then the attacker would not have been provisioned with this key to make this request.

Also, the request if propagated through to the CN, would be denied and the malicious device would be detected as it would not have been authorized for the service by the SM.

6. Inside attacker provides data for services for which it was not authorized

In this scenario, the inside attacker attempts to provide data for services for which it has not been authorized. In this case, there wouldn't be a pending interest for the data on neighboring nodes and they wouldn't have an entry for this service registered for this device in their FIB, so they would not propagate the data. If the service uses a service key, then the inside attacker would not have the required service key to provide the data. Even if the data made it up the subtree to the CN, then the CN would recognize in its FIB that this device node is not authorized to provide the data and potentially cause the SM to revoke the device node and trigger a rekeying of the network forwarding key for the enclave.

7. Inside attacker propagates false routing information

This is a more difficult problem because of the shared network forwarding key for the entire enclave. A malicious node fully registered into the enclave network could propagate false routing information and allow it to essentially perform a denial of service attack against other device nodes. This type of attack is related to the next scenario and proposed mitigation is described below.

8. Inside attacker performs a sinkhole type of attack

In this scenario, the inside attacker can refuse to forward interests or data content through the subtree (blackhole) or selectively forward (greyhole). This could be most effective when combined with the previous attack and advertise fake routes on behalf of other notes making it a root node in the subtree. While our framework doesn't directly address these two attack scenarios, they could be mitigated by including an intrusion detection system (IDS) into the CNs in coordination with the SM to monitor the network topology and activity. Since our network hierarchy relies on a structure consisting of directed acyclic graphs, such as in RPL, an algorithm detection and mitigation scheme could be employed similar to those described in [90] or for wireless sensor networks as in [91][92]. The SM and CM could then detect the malicious node and revoke its keys from their database and FIBs then trigger a rekeying of the network forwarding key for the enclave by forcing nodes to repeat the discovery process and reauthenticate to obtain the new keys.

5.2 Formal Security Verification

In this subsection, I present the results of a formal security verification of my proposed scheme using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. AVISPA is a powerful tool for the automated validation of Internet security-sensitive protocols and applications [93]. It supports a modular and expressive formal language to specify protocols and their security properties. It combines this with modular integration of different back-ends that implement a variety of state-of-the-art automated analysis schemes.

Using AVISPA, I define the protocol steps and security requirements in the High-Level Protocol Specification Language (HLPSL). HLPSL is an expressive, modular, role-based formal language in which you can specify data structures, control flow patterns, alternative intruder models, complex security properties, and different cryptographic primitives with their algebraic properties [93]. These characteristics of HLPSL make it well suited for specifying modern Internet-scale protocols. The specifications written in HLPSL are translated into a rewrite-based formalism *Intermediate Format* (IF). This translation of HLPSL specifications into their equivalent IF specifications is done by the *HLPSL2IF* translator. An IF specification details an infinite-state transition system amenable to formal analysis. These IF specifications are then input to the module back-ends of the AVISPA Tool, which implement different schemes and techniques for analysis.

For my evaluation, I focus on two of these back-ends suited for the protocol presented. The first back-end technique I use is the On-the-fly Model-Checker (OFMC), which performs protocol falsification and bounded verification by exploring the possible transitions described by the IF specification in a demand-driven way. OFMC uses several correct and complete symbolic techniques. It allows for the use of typed and untyped protocol models and the specification of algebraic properties of cryptographic operators [94].

The second AVISPA back-end utilized for my evaluation is the Constraint-Logic-based Attack Searcher (CL-AtSe), which applies constraint solving with powerful simplification heuristics and techniques to eliminate redundancy. It also supports type-

flaw detection and can process the associativity of message concatenation. It takes as input the IF specification and uses rewriting and constraint solving schemes to model all reachable states of the participants to determine if an attack exists with regard to the Dolev-Yao intruder [95]. Dolev-Yao is a formal model to prove the properties of cryptographic protocols [96]. In CL-AtSe, any state-based security property can be modeled, such as secrecy, authentication, fairness etc. Both OFMC and CL-AtSe are well suited for the evaluation of protocols such as those I have presented in this work.

HLPSL is a role-based language, and there are two types of roles: *basic* and *composed* roles. The *basic* roles describe the actions of one agent involved in a single protocol or sub-protocol execution. The *composed* roles instantiate and conjoin one or more other roles. The composed roles include a session role that instantiates the parameters of the basic roles and an environment role, which includes all global variables and defines the protocol sessions. Finally, the security goals of the proposed protocol are defined for the model to check [97].

5.3 Formal Security Verification Results

In my specification, I'm focused on the critical first two stages of the protocol. I defined the basic roles for the new device node, the neighboring device node, and a coordinating node. For simplicity, I combine the roles in our architectural model of the gateway node and cloud security manager into the coordinating node role. This is reasonable for three reasons. First, the coordinating node is considered a trusted device node. Second, the coordinating node, like the gateway and security manager, is

considered unconstrained regarding its processing, storage, and energy capabilities. Finally, these nodes are also assumed to be able to interact using the full NDN stack with greater security capabilities using asymmetric cryptography outside the scope of this work. Our security objectives are to validate the preservation of the secrecy of our symmetric keys and cryptographic primitives against intruders, whether they are resistant to various attacks, and the mutual authentication of the network and the new device node. The full specifications are provided in the appendix. The results of both the OFMC and CL-AtSe back ends verifying the protocol specification are shown in Figure 19. In both cases, the model and state checks indicate the protocols are safe, indicating the keys are secured, and authentication is achieved.

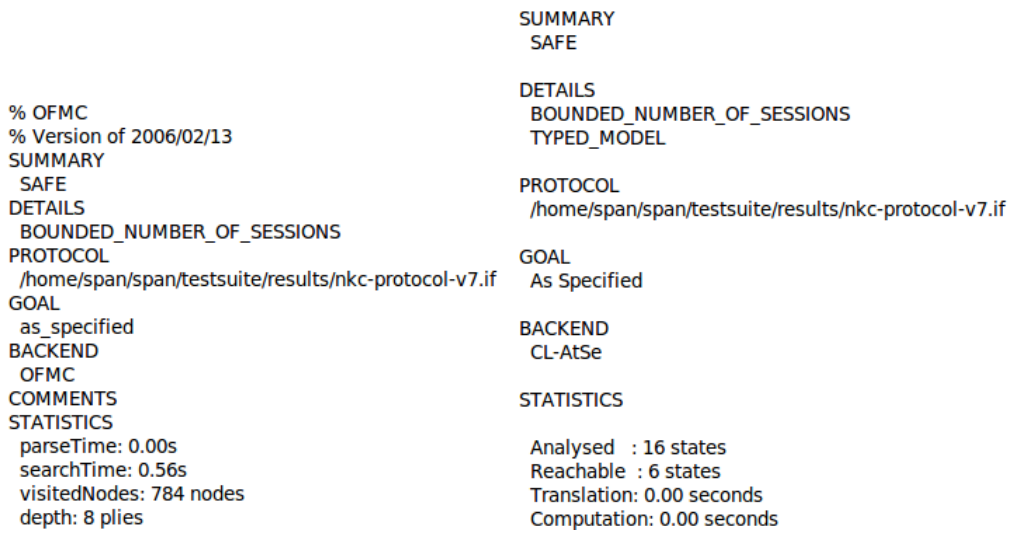

Figure 19. Formal Security Verification Results

5.4 Summary

In this chapter, I described the AVISPA tools and the HLPSL protocol specification language and how they are used to verify the security of an Internet protocol. I described the back-end modules used to verify my protocol OFMC and CL-AtSe. I explain the basic and composition roles considered in my specification and define the security objective. Finally, I show the results of the verification modules. The modules both indicated the protocol was safe.

6. SCALABILITY AND EFFICIENCY EVALUATION

In this section, I discuss a simulation model created to evaluate the scalability and efficiency of the proposed framework and supporting protocols. The objective of this model is to simulate joining many devices into a realistic IoT deployment environment. This is an important aspect of answering part of the initial research questions. To evaluate scalability, I consider the completion time for device nodes to fully integrate into the ICNoT enclave network. I also consider the distance in hop count between the device nodes and their serving coordinating node. To evaluate efficiency, I consider the energy usage, which is best represented by the total transmission burden of each device node to complete the integration. I also consider the size of the subtrees formed within the enclave and demonstrate that this size directly relates to the transmission burden.

The simulation is conducted using ndnSIM, which is a specialized version of the ns-3 simulator for NDN networks [98]. ns-3 is a discrete event simulator that allows the simulation of Internet systems, including the underlying physical and link layers [99]. Simulating large-scale wireless scenarios in ns-3 becomes increasingly difficult as computing interference on the radio channel becomes computationally prohibitive. However, this is not needed because, as described in the network architecture and smart city use case in Chapter 3, it is expected that a smart city will consist of hundreds or thousands of network enclaves, so I need only focus my simulation on a single enclave network. The complete integration in a single enclave is representative of all other enclaves, and the integrations of multiple enclaves can be completed in parallel. This

greatly reduces the complexity and computational expense of the simulation. However, in a smart city environment, there may be multiple CNs deployed per enclave network and hundreds or thousands of enclaves interconnected.

6.1 Simulation Configuration

The simulation was created using an ns-3 extension called ndnSIM [98], which allowed me to model an NDN and employ a custom forwarding strategy, pending interest table (PIT), and forwarding information base (FIB) as described in the protocol. The forwarding strategy encompasses all the forwarding and routing decisions for interests and data sent and received through the enclave subtrees. Additionally, there is a higher level controller implemented to perform the functions within the supporting protocol, such as discover, register, auth, and svcreq. ndnSIM is implemented as a network-layer protocol model and can be run over any available link-layer protocol model. For my scenario, I use an underlying link-layer based on 802.15.4 [32] with 127-byte frames, similar to that used in the analytical evaluation of [60]. I use ndnSIM's Low-Rate Wireless Personal Area Network Device model using slotted carrier-sense multiple access with collision avoidance (CSMA/CA) combined with Constant Speed Propagation Delay and Log Distance Propagation Loss models in the signal channel. This was chosen because it represents a realistic IoT environment at the network edge. Because 802.15.4 does not support packet fragmentation, I implement an abstraction layer on top of it through the ndn::NetDeviceFace to allow for hop-by-hop fragmentation and reassembly

of the NDN interests and data packets. For passing interest parameters in our lightweight NDN format, I use the payload field of the ndnSIM interest object.

6.1.1 Simulation Inputs and Design

The scenario considers four arguments as inputs to the simulation, consisting of:

1. A random seed is used for the pseudo-random number generator affecting the topology and placement of device nodes within the simulation area.
2. A run seed is used for all other pseudo-random number generation for the rest of the scenario, such as network behavior.
3. The number of nodes to place in the simulation area.
4. The size of the simulation area as a square area. The input is in meters, so the area of input n will be n m^2 x n m^2.

The first seed allows to rerun a scenario with the same topography but altering the network performance. The second seed us used to randomize all other aspects of the simulation, including network and radio channel behaviors. The last two inputs control the number of devices nodes and the area in which they are deployed.

Figure 20 shows an example distribution of 100 device nodes placed on a $100m^2$ by $100m^2$ area (0.01 square km). The coordinating node for this enclave is in the middle, represented by a blue dot. Additional device nodes are added in a uniform random distribution within the area. To make the deployment more realistic, the device nodes are activated over an exponential distribution of $\lambda^{-1} = 120s$ to begin integrating into the enclave network.

This constitutes an enclave equivalent to 0.01km^2. In the context of a smart city, with multiple coordinating nodes and enclaves, there would be hundreds of thousands of device nodes per km^2. For this simulation, I consider placing the coordinating node in the center for all simulation runs. In practice, there may be other considerations. As my results will demonstrate, careful placement of the coordinating node should be considered in real-world deployments to minimize obstructions and maximize the number of device nodes that can communicate directly with the coordinating node. It is desirable to minimize the length of subtrees formed among the device nodes in the enclave. As results will demonstrate that longer subtrees increase the transmission burden of the device. nodes.

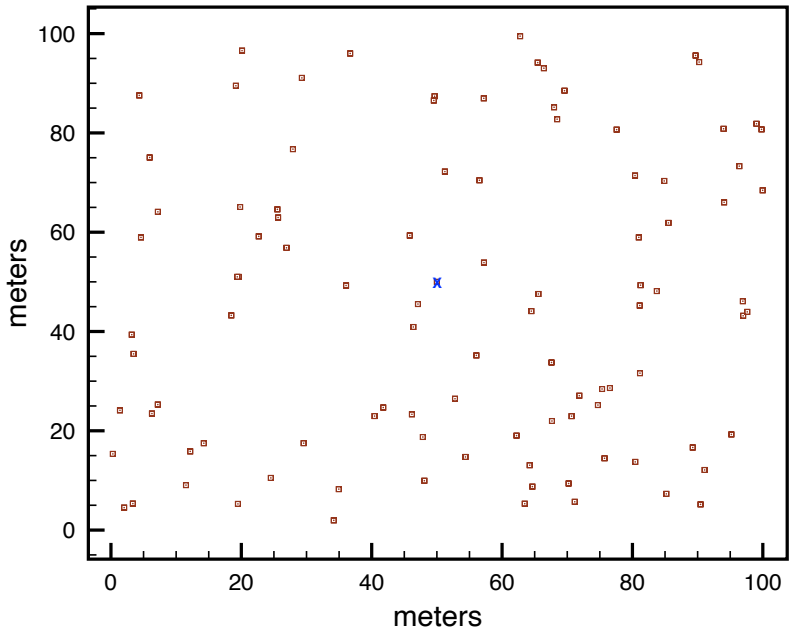

Figure 20. Example of device node distribution

6.1.2 Simulation Outputs

Using ndnSIM, I built in a logging module to the simulation to capture data for each node, including:

1. Time until the device node is fully integrated
2. Device node ID
3. The next-hop ID of the next device node, 0 if it is the coordinating node
4. The coordinating node the device node is associated with
5. The length of the subtree in hops from the device node's coordinating node
6. The position of the device node on the area of deployment (x,y,z)
7. The total amount of data transmitted by the device node

6.2 Simulation Results

6.2.1 Simulation Results Introduction

First, I am primarily interested in the time required for an increasing number of nodes to be fully registered, secure forwarding established, and a single service authorized within an enclave network of a fixed size. I do not account for the presence of application data that may exist. I am considering a new deployment of different numbers of device nodes into an IoT enclave, which simulates two different densities of devices.

My model allows us to input the number of nodes along with the size of the area in which they are deployed, represented in meters squared by meters squared. This will be the geospatial size of the enclave.

6.2.2 Simulation in Fixed Area

In Figure 21, I executed over 100 runs with random topographies and random seeds, which affect device location, activation times, network, and radio channel behaviors. I show the average results for two densities of device nodes: 50 nodes and 100 nodes. They are placed within a fixed area of 100m by $100m^2$ or 0.01 km^2. This would represent densities of up to 10,000 device nodes per km^2. On the x-axis is the time in seconds, and on the y-axis is the percentage of device nodes that have fully integrated into the enclave network, having completed the four stages.

As the devices are deployed and activated, they automatically begin discovery and registration. There is an obvious but slight increase in the time required for all devices to fully integrate into the enclave network as the density increases, indicating that the increase in radio interference among devices using the same channel causes retransmissions that delay the process. The results indicate a promising level of manageability and scalability as the time to complete integration is around four minutes for the 50 device node deployment and only slightly higher for the denser deployment of 100 nodes. This density would support tens of thousands of devices per square kilometer with multiple enclaves. By using non-overlapping channels on multiple coordinating nodes in each enclave, it could support hundreds of thousands of device nodes per square

kilometer. The worst-case time for full integration out of 100 runs was 331.3 seconds, about 5 and a half minutes, which is not much greater than the average case for 100 nodes at 4 minutes and 15 seconds. This result is reasonable considering the process will typically only be done once and compares favorably to results in other work [60].

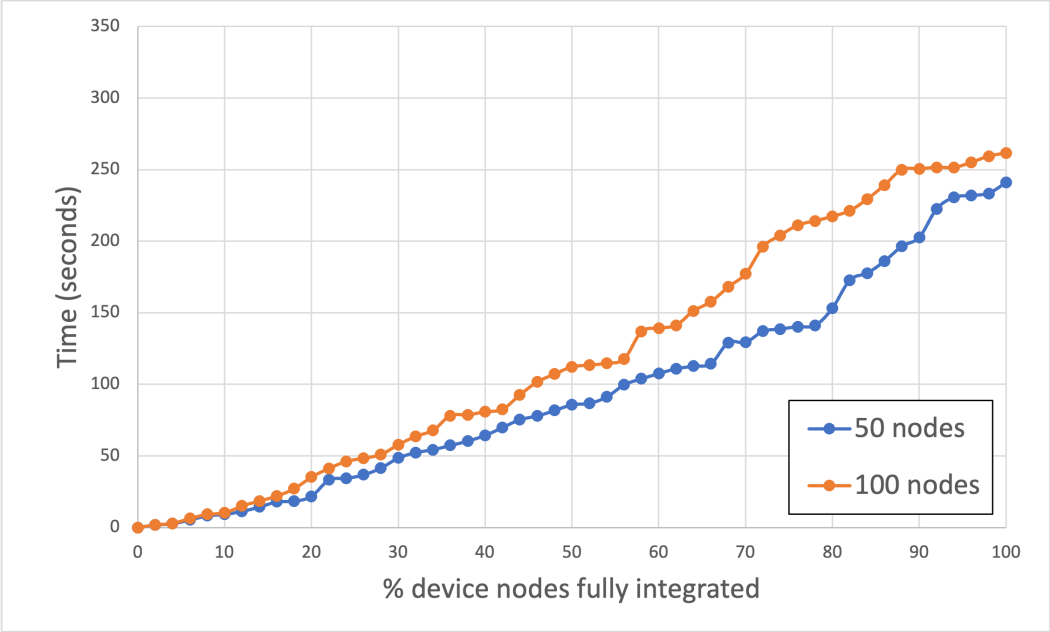

Figure 21. Average time for completion of device node integration

Next, I consider the size of the subtrees formed in the enclave. As the next section will demonstrate, the size of the subtree has a direct relationship to the number of transmissions by the device nodes because they must propagate data and interests from their constituent nodes. Figure 22 shows the percentage of nodes across the simulation executions by subtree size. Across all executions, most device nodes communicate

directly with their coordinating node. On average, 81.7% of device nodes were paired directly with the CN and had no children. These device nodes would have the optimal minimum transmission burden. The hop count distance from the coordinating node corresponds logically with the subtree size. As is visible in Figure 22, as I increase the density of deployment, there is a tendency to increase the likelihood of the presence of longer subtrees even though the physical distance between the device nodes and coordinating nodes was the same. This is related to the increasing density causing increased interference, reducing the effective transmission range of the device nodes. In real-world practice, large subtrees might be averted by strategic placement of the coordinating nodes relative to the device nodes and consideration of obstructions.

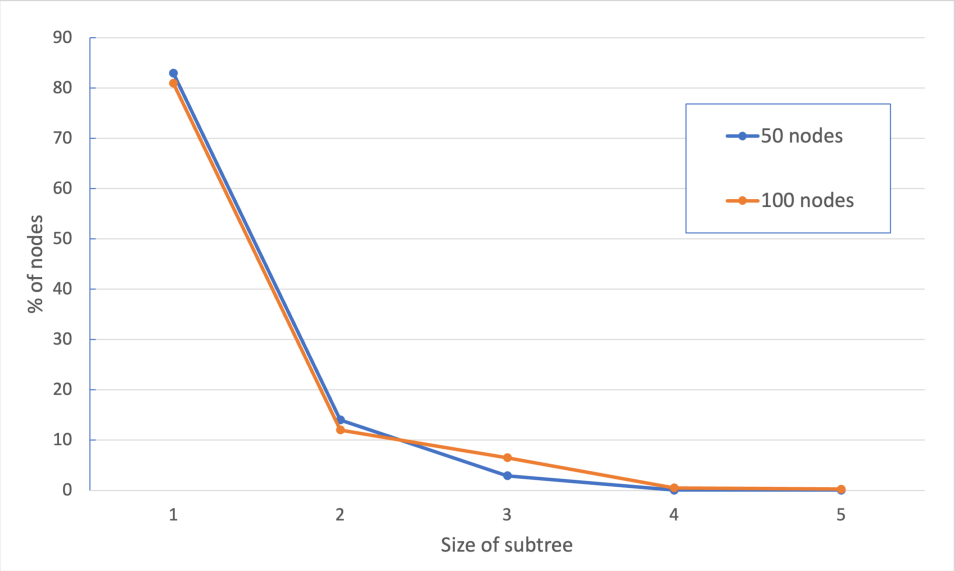

Figure 22. Percentage of nodes with subtree size

Next, I consider the transmission burden by the device nodes related to the size of their subtrees. In these IoT enclave networks, efficiency can be evaluated by considering the amount of energy expended, which is primarily from wireless transmissions by the device nodes. Figure 23 shows the subtree sizes on the x-axis in relation to the total transmission burden in KiB for each subtree. Devices with a subtree size of 0 are directly communicating with their coordinating node. Here there is a clear relationship between the transmission burden and the number of devices in a subtree. As the subtree size increases, so does the amount of transmitted data. And as the previous figure demonstrated, at the larger density of device nodes deployed, longer subtrees form with an increased transmission burden. This simulation only considers traffic for the data and interests in the protocol and does not consider any application traffic. This simulation only considers deployment in a new environment without existing device nodes already present and integrated into the enclave network. Future work may consider deployment in an environment with existing device nodes and simulate different levels of application traffic.

A single device node on its own may transmit about 427 bytes, the equivalent of four full 802.15.4 packets. However, in practice, it transmits several smaller packets across the four exchanges with a coordinating node. For most device nodes, the average total amount transmitted is under 8 KiB to complete integration. This is larger because with many nodes activating, the discover interest must be broadcast, which multiplies the number of nodes in the path that may receive and process them. While this is reasonable

and efficient, the placement of coordinating nodes and the density of devices deployed affects the size of the subtrees formed, and with larger subtrees, there is a larger transmission burden.

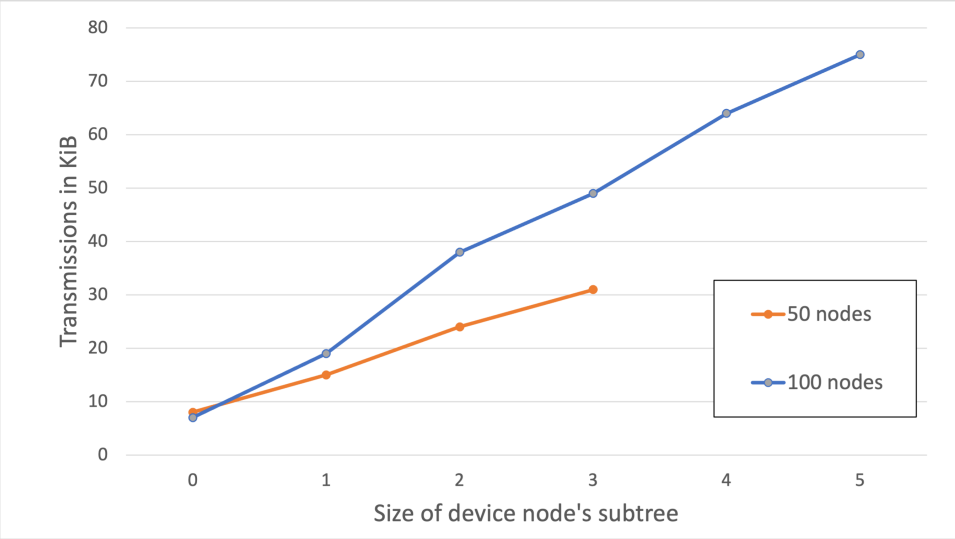

Figure 23. Transmission by subtree size

6.2.3 Simulation in Different Densities

In Figure 24, I consider the placement of a fixed number of device nodes in different densities by adjusting the size of the placement area. I consider the same metrics that were used in Figure 21. In this scenario, I consider the time for 100 device nodes to complete integration into the IoT enclave at densities of 100m x 100m^2 (0.01 km^2), 200m x 200m^2 (0.04 km^2), and 500m x 500m^2 (0.25 km^2). The results demonstrate that by

increasing the sparseness, it increases the probability that device nodes will form longer subtrees and have a greater hop count to the coordinating node in the enclave.

The x-axis shows the percentage of devices fully integrated, while the y-axis shows the time in seconds to completion. The trends are similar to what was shown in Figure 21 with 100 device nodes in a 0.01 km^2 area, which is the densest area and has the slowest time for complete integration. This is because, as the density increases, there is a greater number of retransmissions needed due to interference on the radio channel. Notably, the sparsest scenario at .25 square kilometers has a slower initial progression until a few minutes pass when the integration time begins to level off. This suggests poor connectivity until enough device nodes are online and integrated into the enclave network. After enough device nodes (over 50%) are active, they can then serve as intermediate nodes to the new activating devices allowing them to reach the coordinating nodes. The sparsity of the deployment means less interference, requiring fewer retransmissions. In all cases, complete integration is achieved in around 4 minutes, with the sparsest deployment performing best.

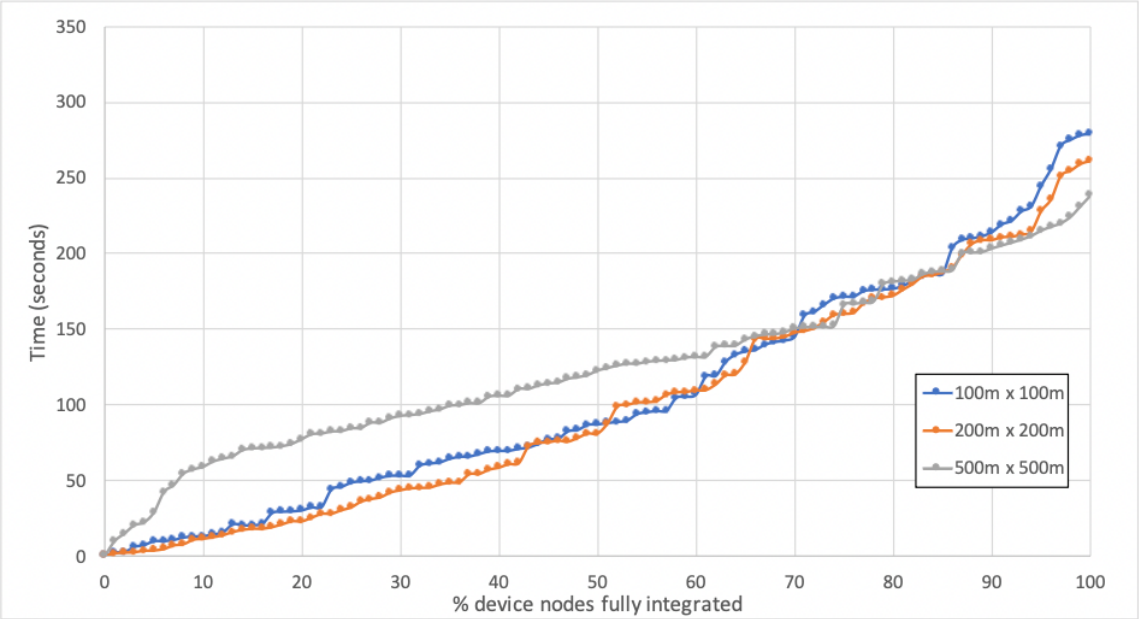

Figure 24. Device node integration completion times at different densities

Finally, in Figure 25, I consider transmission burden by subtree size in different densities with a fixed number of 100 device nodes deployed. Results are similar to Figure 23, where higher densities of devices tend to have a higher transmission burden caused by retransmissions due to interference. Also, it is notable that the least dense deployment tends to have longer subtrees, leading to a larger overall transmission burden.

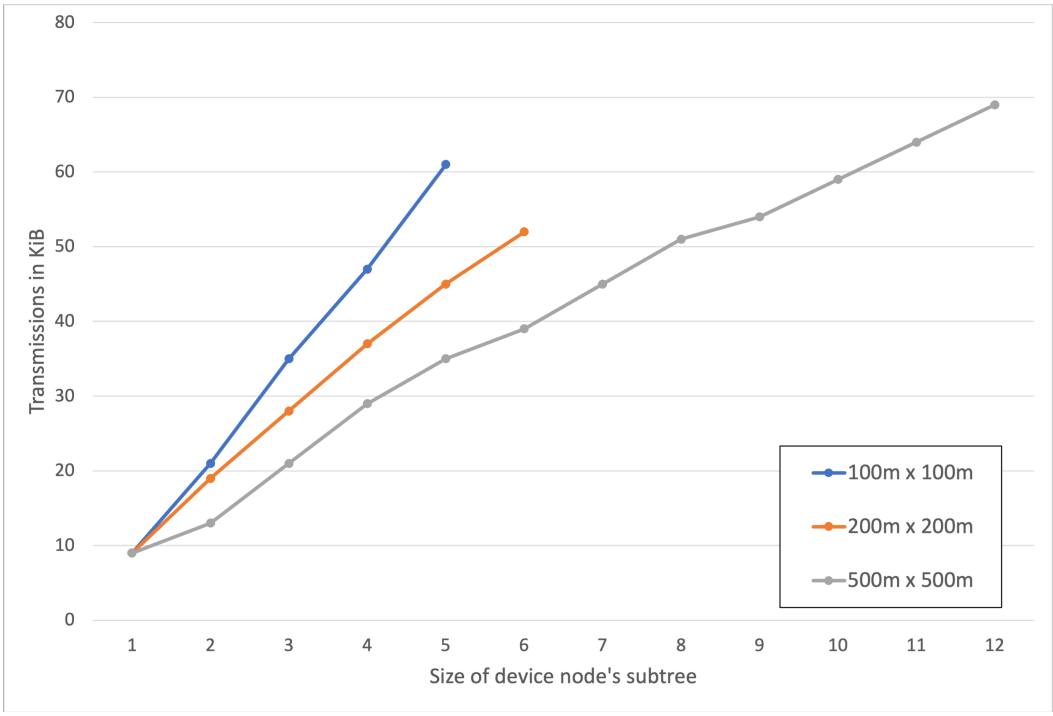

Figure 25. Transmission burden by subtree size

In addition to the simulation parameters demonstrated and discussed in the figures, executions were run with increasing densities up to 200 device nodes within 50 square meters, but in this scenario, only about half the devices were able to complete all stages, and the network routing was unable to converge due to interference on the radio channel. To overcome this, multiple CNs could be deployed, operating on separate channels. Alternatively, the devices could reduce power output, which could increase the length of the device subtrees, but reduce overall interference.

6.4 Summary

The simulation results indicate that the proposed work is both scalable and efficient in that it supports a large number of devices fully integrating into an IoT enclave network in both a reasonable time and using minimal transmission burden and energy. By combining multiple enclaves, it could easily scale to 10,000 devices per square kilometer. Additional simulations indicated densities of up to 80,000 devices per square kilometer with similar efficiency. Deploying additional coordinating nodes on non-overlapping channels could support densities on the order of hundreds of thousands of devices per square kilometer. This would effectively support deployment in use cases such as smart campuses, grids, or cities.

7. CONCLUSIONS

In this work, I present a framework and supporting protocols that address the authentication, registration, secure forwarding, and service authorization and discovery of constrained devices into an ICN-based IoT in a way that is highly scalable and efficient. I leverage a mesh network with a hierarchical structure to enhance scalability. The device nodes participating in our architecture are assumed to be constrained, so cryptographic operations are kept to a minimum using lightweight symmetric encryption functions while they rely on unconstrained coordinating nodes in concert with a security manager service to manage authentication, key distribution. and offload computation from the devices.

My work provides a way for constrained devices to fully integrate into an ICN-based IoT securely and efficiently. This includes not just authentication and secure routing but also service authorization and discovery, using an architecture that can scale for use in a smart campus or city. This framework provides capabilities for protecting against external attackers. I then expand its security capabilities to protect against inside attackers by establishing mechanisms to establish and monitor the trust of devices based on their behavior and quality of service attributes over time. I also show how to integrate this capability to make encryption-based access control decisions based on this trust. A simulation model was used to demonstrate the scalability and efficiency of the proposed architecture, framework, and supporting protocols.

Future work involves implementing a functional representation of the architecture in software to demonstrate the protocol using a set of test data generated by enhancing the trust model in an existing simulation of our previous work. This could involve developing an implementation in RIOT OS or some other constrained IoT operating system [100]. I also propose to separate the concept of the coordinator nodes as semi-trusted and introduce the concept of edge network support nodes (NSD) that might include local routers, switches, bridges, or other network-level equipment that support the ICN and consider them semi-trusted devices to protect against additional attack scenarios in a more realistic edge network environment. Additionally, the simulation model could be expanded to consider different placements of the coordinating nodes and the presence of existing device nodes already integrated into the network with application data present.

8. REFERENCES

[1] S. Akhshabi and C. Dovrolis, "The evolution of layered protocol stacks leads to an hourglass-shaped architecture," in *Proceedings of the ACM SIGCOMM 2011 conference*, New York, NY, USA, Aug. 2011, pp. 206–217. doi: 10.1145/2018436.2018460.

[2] "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," p. 38, 2018.

[3] A. Djama, B. Djamaa, and M. R. Senouci, "Information-Centric Networking solutions for the Internet of Things: A systematic mapping review," *Computer Communications*, vol. 159, pp. 37–59, Jun. 2020, doi: 10.1016/j.comcom.2020.05.003.

[4] J. Rexford and C. Dovrolis, "Future Internet architecture: clean-slate versus evolutionary research," *Commun. ACM*, vol. 53, no. 9, pp. 36–40, Sep. 2010, doi: 10.1145/1810891.1810906.

[5] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, Jul. 2011, doi: 10.1109/MCOM.2011.5936152.

[6] L. Zhang *et al.*, "Named Data Networking (NDN) Project," *Relatorio Tecnico NDN-0001, Xerox Palo Alto Research Center-PARC*, vol. 157, p. 158, 2010.

[7] T. Berners-Lee, L. M. Masinter, and M. P. McCahill, "Uniform Resource Locators (URL)," Internet Engineering Task Force, Request for Comments RFC 1738, Dec. 1994. doi: 10.17487/RFC1738.

[8] A. Ghodsi, B. Ohlman, J. Ott, I. Solis, and M. Wählisch, "Information-centric networking – Ready for the real world (Dagstuhl Seminar 12361)," *Dagstuhl Reports*, vol. 2, no. 9, pp. 1–14, 2013, doi: 10.4230/DagRep.2.9.1.

[9] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent Advances in Information-Centric Networking-Based Internet of Things (ICN-IoT)," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2128–2158, Sep. 2018, doi: 10.1109/JIOT.2018.2873343.

[10] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the IoT: experiments with NDN in the wild," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, New York, NY, USA, Sep. 2014, pp. 77–86. doi: 10.1145/2660129.2660144.

[11] J. M. Hernández-Muñoz *et al.*, "Smart Cities at the Forefront of the Future Internet," in *The Future Internet*, Berlin, Heidelberg, 2011, pp. 447–462. doi: 10.1007/978-3-642-20898-0_32.

[12] J. A. Cordero, J. Yi, T. Clausen, and E. Baccelli, "Enabling Multihop Communication in Spontaneous Wireless Networks," in *Recent Advances in Networking 1*, ACM SIGCOMM eBook, 2013, pp. 413–457.

[13] "ZigBee Pro Specification." The Zigbee Alliance, Aug. 2015.

[14] G. Mulligan, “The 6LoWPAN architecture,” in *Proceedings of the 4th workshop on Embedded networked sensors*, New York, NY, USA, Jun. 2007, pp. 78–82. doi: 10.1145/1278972.1278992.
[15] R. Alexander *et al.*, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” Internet Engineering Task Force, Request for Comments RFC 6550, Mar. 2012. doi: 10.17487/RFC6550.
[16] K. Ashton, “That ‘Internet of Things’ Thing. In the real world, things matter more than ideas.,” *RFID Journal*, Jun. 22, 2009. Accessed: Apr. 06, 2017. [Online]. Available: http://www.rfidjournal.com/articles/view?4986
[17] K. Evangelos A., T. Nikolaos D., and B. Anthony C., “Integrating RFIDs and Smart Objects into a UnifiedInternet of Things Architecture,” *Advances in Internet of Things*, vol. 2011, Apr. 2011, doi: 10.4236/ait.2011.11002.
[18] R. Want, “An introduction to RFID technology,” *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, Jan. 2006, doi: 10.1109/MPRV.2006.2.
[19] D. L. Brock, “The Electronic Product Code (EPC),” p. 21.
[20] C. F. García-Hernández, P. H. Ibargüengoytia-González, J. García-Hernández, and J. A. Pérez-Díaz, “Wireless Sensor Networks and Applications: a Survey,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 7.3, pp. 264–273, 2007.
[21] F. L. Lewis, “Wireless Sensor Networks,” *Smart Environments: Technologies, Protocols, and Applications*, p. 19, 2004.
[22] R. van Kranenburg, *The Internet of Things: A Critique of Ambient Technology and the All-seeing Network of RFID*. Institute of Network Cultures, 2008.
[23] J. Wan, D. Li, C. Zou, and K. Zhou, “M2M Communications for Smart City: An Event-Based Architecture,” in *2012 IEEE 12th International Conference on Computer and Information Technology*, Oct. 2012, pp. 895–900. doi: 10.1109/CIT.2012.188.
[24] Y. Ben Saied, A. Olivereau, and M. Laurent, “A Distributed Approach for Secure M2M Communications,” in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, May 2012, pp. 1–7. doi: 10.1109/NTMS.2012.6208702.
[25] J. Wan, M. Chen, F. Xia, L. Di, and K. Zhou, “From machine-to-machine communications towards cyber-physical systems,” *Computer Science and Information Systems*, vol. 10, no. 3, pp. 1105–1128, 2013.
[26] M. Chen, J. Wan, and F. Li, “Machine-to-Machine Communications: Architectures, Standards and Applications,” *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 6, no. 2, pp. 480–497, 2012, doi: 10.3837/tiis.2012.02.002.
[27] S. Karnouskos, “Cyber-Physical Systems in the SmartGrid,” in *2011 9th IEEE International Conference on Industrial Informatics*, Jul. 2011, pp. 20–23. doi: 10.1109/INDIN.2011.6034829.
[28] P. Friess *et al.*, “Europe’s IoT Strategic Research Agenda 2012,” 2012, pp. 22–117.
[29] M. R. Palattella *et al.*, “Standardized Protocol Stack for the Internet of (Important) Things,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013, doi: 10.1109/SURV.2012.111412.00158.

[30] J. Postel, “RFC 793 - Transmission Control Protocol,” RFC 793, 1981.
[31] I. Mohammad, “Cisco live! Partner Enablement for building IoT Endpoints and Applications,” Barcelona, Spain, Feb. 2018. [Online]. Available: https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2018/pdf/BRKIOT-1390.pdf
[32] “IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),” *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011, doi: 10.1109/ieeestd.2011.6012487.
[33] “IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer,” *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, Apr. 2012, doi: 10.1109/IEEESTD.2012.6185525.
[34] “IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 2: Alternative Physical Layer Extension to support one or more of the Chinese 314-316 MHz, 430-434 MHz, and 779-787 MHz bands,” IEEE.
[35] “IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. Amendment 3: Specifications for Operation in Additional Regulatory Domains,” IEEE.
[36] “Zigbee specification.” ZigBee Alliance, 2006.
[37] “ZigBee 2007 and ZigBee Pro,” Elsevier, 2008, pp. 389–403. doi: 10.1016/b978-0-7506-8597-9.00013-6.
[38] “ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: Process control and related applications,” *isa.org*. https://www.isa.org/products/ansi-isa-100-11a-2011-wireless-systems-for-industr (accessed May 09, 2022).
[39] T. Lennvall, S. Svensson, and F. Hekland, “A comparison of WirelessHART and ZigBee for industrial applications,” 2008, pp. 85–88. doi: 10.1109/wfcs.2008.4638746.
[40] “Z-Wave Networking Basics,” Aug. 2016.
[41] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, “A concrete security treatment of symmetric encryption,” in *Proceedings 38th Annual Symposium on Foundations of Computer Science*, Oct. 1997, pp. 394–403. doi: 10.1109/SFCS.1997.646128.
[42] G. Montenegro, C. Schumacher, and N. Kushalnagar, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” Internet Engineering Task Force, Request for Comments RFC 4919, Aug. 2007. doi: 10.17487/RFC4919.
[43] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, “RFC 4944 - Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” RFC 4944, 2007.

[44] J. Hui and P. Thubert, “RFC 6282 - Compression Format for IPv6 Datagrams Over IEEE 802.15. 4-based Networks,” RFC 6282, Sep. 2011.
[45] E. Kim, D. Kaspar, and J. P. Vasseur, “Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs),” Internet Engineering Task Force, Request for Comments RFC 6568, Apr. 2012. doi: 10.17487/RFC6568.
[46] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” Internet Engineering Task Force, Request for Comments RFC 7252, Jun. 2014. doi: 10.17487/RFC7252.
[47] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security Version 1.2,” Internet Engineering Task Force, Request for Comments RFC 6347, Jan. 2012. doi: 10.17487/RFC6347.
[48] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, “openPDS: Protecting the Privacy of Metadata through SafeAnswers,” *PLOS ONE*, vol. 9, no. 7, p. e98790, Jul. 2014, doi: 10.1371/journal.pone.0098790.
[49] A. Arabo, I. Brown, and F. El-Moussa, “Privacy in the Age of Mobility and Smart Devices in Smart Homes,” in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, Sep. 2012, pp. 819–826. doi: 10.1109/SocialCom-PASSAT.2012.108.
[50] H. Gross, M. Hölbl, D. Slamanig, and R. Spreitzer, “Privacy-Aware Authentication in the Internet of Things,” in *Cryptology and Network Security*, Cham, 2015, pp. 32–39. doi: 10.1007/978-3-319-26823-1_3.
[51] A. Chakravorty, T. Wlodarczyk, and C. Rong, “Privacy Preserving Data Analytics for Smart Homes,” in *2013 IEEE Security and Privacy Workshops*, May 2013, pp. 23–27. doi: 10.1109/SPW.2013.22.
[52] A. F. Skarmeta, J. L. Hernández-Ramos, and M. V. Moreno, “A decentralized approach for security and privacy challenges in the Internet of Things,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 67–72. doi: 10.1109/WF-IoT.2014.6803122.
[53] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long, “Managing flash crowds on the Internet,” in *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003.*, Oct. 2003, pp. 246–249. doi: 10.1109/MASCOT.2003.1240667.
[54] “A delay-tolerant network architecture for challenged internets | Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications.” https://dl.acm.org/doi/abs/10.1145/863955.863960 (accessed Jul. 02, 2022).
[55] B. Nour *et al.*, “A survey of Internet of Things communication using ICN: A use case perspective,” *Computer Communications*, vol. 142–143, pp. 95–123, Jun. 2019, doi: 10.1016/j.comcom.2019.05.010.
[56] R. Ravindran, T. Biswas, X. Zhang, A. Chakraborti, and G. Wang, “Information-centric networking based homenet,” in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 1102–1108.

[57] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang, “Securing building management systems using named data networking,” *IEEE Network*, vol. 28, no. 3, pp. 50–56, May 2014, doi: 10.1109/MNET.2014.6843232.
[58] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, “Information Centric Networking in IoT scenarios: The case of a smart home,” in *2015 IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 648–653. doi: 10.1109/ICC.2015.7248395.
[59] M. Enguehard, R. Droms, and D. Rossi, “On the Cost of Secure Association of Information Centric Things,” in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, New York, NY, USA, Sep. 2016, pp. 207–208. doi: 10.1145/2984356.2985237.
[60] A. Compagno, M. Conti, and R. Droms, “OnboardICNg: a Secure Protocol for On-boarding IoT Devices in ICN,” in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, New York, NY, USA, Sep. 2016, pp. 166–175. doi: 10.1145/2984356.2984374.
[61] T. Mick, R. Tourani, and S. Misra, “LASeR: Lightweight Authentication and Secured Routing for NDN IoT in Smart Cities,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 755–764, Apr. 2018, doi: 10.1109/JIOT.2017.2725238.
[62] “A Quick Oerview of NDN-Lite.” https://ndn-lite.named-data.net/1-overview.html (accessed May 11, 2022).
[63] A. Aboodi, T.-C. Wan, and G.-C. Sodhy, “Survey on the Incorporation of NDN/CCN in IoT,” *IEEE Access*, vol. 7, pp. 71827–71858, 2019, doi: 10.1109/ACCESS.2019.2919534.
[64] X. Sun and N. Ansari, “EdgeIoT: Mobile Edge Computing for the Internet of Things,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, Dec. 2016, doi: 10.1109/MCOM.2016.1600492CM.
[65] K. Moriarty, B. Kaliski, and A. Rusch, “PKCS #5: Password-Based Cryptography Specification Version 2.1,” Internet Engineering Task Force, Request for Comments RFC 8018, Jan. 2017. doi: 10.17487/RFC8018.
[66] C. Percival and S. Josefsson, “The scrypt Password-Based Key Derivation Function,” Internet Engineering Task Force, Request for Comments RFC 7914, Aug. 2016. doi: 10.17487/RFC7914.
[67] I. T. L. Computer Security Division, “Lightweight Cryptography | CSRC | CSRC,” *CSRC | NIST*, Jun. 02, 2022. https://csrc.nist.gov/projects/lightweight-cryptography (accessed Aug. 03, 2022).
[68] N. Clark, “ICN Suitability for the Internet of Things,” in *IADIS International Conference e-Society 2020*, 2020, pp. 167–170.
[69] N. K. Clark, “Securely & Efficiently Integrating Constrained Devices into an ICN-IoT,” in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, Jun. 2021, pp. 536–541. doi: 10.1109/WF-IoT51360.2021.9595708.
[70] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, “Secure Surveillance Framework for IoT Systems Using Probabilistic Image Encryption,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3679–3689, Aug. 2018, doi: 10.1109/TII.2018.2791944.

[71] M. Alhanahnah, P. Bertok, Z. Tari, and S. Alouneh, “Context-Aware Multifaceted Trust Framework For Evaluating Trustworthiness of Cloud Providers,” *Future Generation Computer Systems*, vol. 79, pp. 488–499, Feb. 2018, doi: 10.1016/j.future.2017.09.071.
[72] R. W. Saaty, “The analytic hierarchy process—what it is and how it is used,” *Mathematical Modelling*, vol. 9, no. 3, pp. 161–176, Jan. 1987, doi: 10.1016/0270-0255(87)90473-8.
[73] D. Soukup, O. Hujňák, S. Štefunko, R. Krejčí, and E. Grešák, “Security Framework for IoT and Fog Computing Networks,” in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Dec. 2019, pp. 87–92. doi: 10.1109/I-SMAC47947.2019.9032592.
[74] G. George and S. M. Thampi, “A Graph-Based Security Framework for Securing Industrial IoT Networks From Vulnerability Exploitations,” *IEEE Access*, vol. 6, pp. 43586–43601, 2018, doi: 10.1109/ACCESS.2018.2863244.
[75] F. Al-Turjman, Y. K. Ever, E. Ever, H. X. Nguyen, and D. B. David, “Seamless Key Agreement Framework for Mobile-Sink in IoT Based Cloud-Centric Secured Public Safety Sensor Networks,” *IEEE Access*, vol. 5, pp. 24617–24631, 2017, doi: 10.1109/ACCESS.2017.2766090.
[76] J. Liang, M. Zhang, and V. C. M. Leung, “A Reliable Trust Computing Mechanism Based on Multisource Feedback and Fog Computing in Social Sensor Cloud,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5481–5490, Jun. 2020, doi: 10.1109/JIOT.2020.2981005.
[77] L. Lu and Y. Yuan, “A novel TOPSIS evaluation scheme for cloud service trustworthiness combining objective and subjective aspects,” *Journal of Systems and Software*, vol. 143, pp. 71–86, Sep. 2018, doi: 10.1016/j.jss.2018.05.004.
[78] I. García-Magariño, S. Sendra, R. Lacuesta, and J. Lloret, “Security in Vehicles With IoT by Prioritization Rules, Vehicle Certificates, and Trust Management,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5927–5934, Aug. 2019, doi: 10.1109/JIOT.2018.2871255.
[79] B. Nour, K. Sharif, F. Li, and Y. Wang, “Security and Privacy Challenges in Information-Centric Wireless Internet of Things Networks,” *IEEE Security & Privacy*, vol. 18, no. 2, pp. 35–45, Mar. 2020, doi: 10.1109/MSEC.2019.2925337.
[80] K. Nichols, “Trust schemas and ICN: key to secure home IoT,” in *Proceedings of the 8th ACM Conference on Information-Centric Networking*, New York, NY, USA, Sep. 2021, pp. 95–106. doi: 10.1145/3460417.3482972.
[81] D. Boneh and M. Franklin, “Identity-Based Encryption from the Weil Pairing,” *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, Jan. 2003, doi: 10.1137/S0097539701398521.
[82] A. Sahai and B. Waters, “Fuzzy Identity-Based Encryption | SpringerLink,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2005, pp. 457–473.
[83] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, “CP-ABE With Constant-Size Keys for Lightweight Devices,” *IEEE Transactions on Information*

Forensics and Security, vol. 9, no. 5, pp. 763–771, May 2014, doi: 10.1109/TIFS.2014.2309858.
[84] V. Odelu and A. K. Das, "Design of a new CP-ABE with constant-size secret keys for lightweight devices using elliptic curve cryptography," *Security and Communication Networks*, vol. 9, no. 17, pp. 4048–4059, 2016, doi: 10.1002/sec.1587.
[85] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Lightweight elliptic curve cryptography accelerator for internet of things applications," *Ad Hoc Networks*, vol. 103, p. 102159, Jun. 2020, doi: 10.1016/j.adhoc.2020.102159.
[86] B. Nour, H. Khelifi, R. Hussain, S. Mastorakis, and H. Moungla, "Access Control Mechanisms in Named Data Networks: A Comprehensive Survey," *ACM Computing Surveys*, 2020.
[87] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, May 2003, pp. 113–127. doi: 10.1109/SNPA.2003.1203362.
[88] K. Prathapchandran and T. Janani, "A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest – RFTRUST," *Computer Networks*, vol. 198, p. 108413, Oct. 2021, doi: 10.1016/j.comnet.2021.108413.
[89] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Feb. 2017, pp. 32–37. doi: 10.1109/I-SMAC.2017.8058363.
[90] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing Attacks and Mitigation Methods for RPL-Based Internet of Things," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1582–1606, 2019, doi: 10.1109/COMST.2018.2885894.
[91] E. C. H. Ngai, J. Liu, and M. R. Lyu, "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks," in *2006 IEEE International Conference on Communications*, Jun. 2006, vol. 8, pp. 3383–3389. doi: 10.1109/ICC.2006.255595.
[92] H. Shafiei, A. Khonsari, H. Derakhshi, and P. Mousavi, "Detection and mitigation of sinkhole attacks in wireless sensor networks," *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 644–653, May 2014, doi: 10.1016/j.jcss.2013.06.016.
[93] A. Armando *et al.*, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *Computer Aided Verification*, Berlin, Heidelberg, 2005, pp. 281–285. doi: 10.1007/11513988_27.
[94] D. Basin, S. Mödersheim, and L. Viganò, "OFMC: A symbolic model checker for security protocols," *Int J Inf Secur*, vol. 4, no. 3, pp. 181–208, Jun. 2005, doi: 10.1007/s10207-004-0055-7.
[95] M. Turuani, "The CL-Atse Protocol Analyser," in *Term Rewriting and Applications*, Berlin, Heidelberg, 2006, pp. 277–286. doi: 10.1007/11805618_21.
[96] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983, doi: 10.1109/TIT.1983.1056650.

[97] Y. Chevalier *et al.*, “A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols,” 2004, p. 13 p. Accessed: Aug. 01, 2022. [Online]. Available: https://hal.inria.fr/inria-00099882
[98] A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM: ndn simulator for NS-3,” Jan. 2012.
[99] G. F. Riley and T. R. Henderson, “The ns-3 Network Simulator,” in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds. Berlin, Heidelberg: Springer, 2010, pp. 15–34. doi: 10.1007/978-3-642-12331-3_2.
[100] “RIOT - The friendly Operating System for the Internet of Things.” https://www.riot-os.org/ (accessed Aug. 03, 2022).

BIOGRAPHY

Nicholas K. Clark received his Bachelor of Science degree in Information Technology and Computer Science in 2009 from George Mason University. He subsequently received his Master of Science in Information Security and Assurance in 2010 and graduate certificates in Computer Networking and Software Engineering, also from George Mason University, in 2011 and 2013.